질의 사항

1. 역인덱스에 Document의 count 저장 여부
2. AWS ES는 Domain이 생성된 이후에 인스턴스 type이나 수의 자유성
3. Data를 분할하여 Migration 가능 여부
4. 다른 Version의 ES 끼리 Data Migration 방법
5. ES의 Data를 csv 혹은 Json으로 Export하는 방법

---

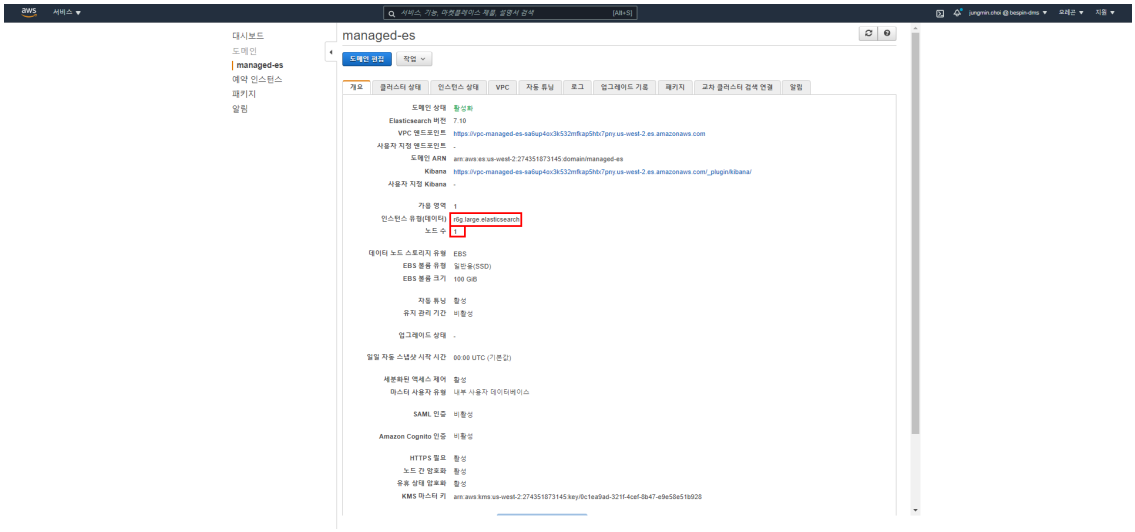
1. 역인덱스에 Document의 count 저장 여부

아닙니다. 역인덱스 , term에 대한 count도 조회는 가능하나 count api를 활용해야만 가능합니다.
즉, 해당 term에 대한 counting을 다시 수행해야 합니다.

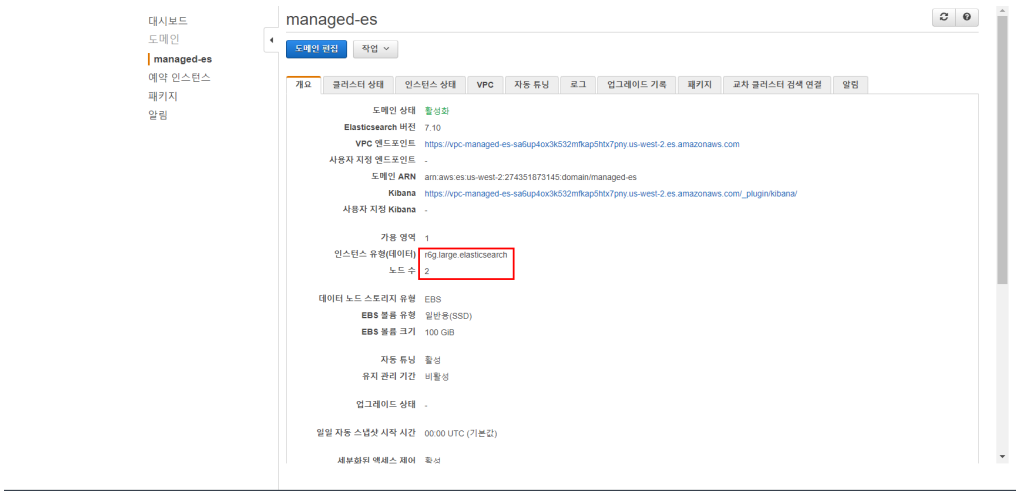
참고 URL :
https://stackoverflow.com/questions/44590350/finding-the-number-of-documents-that-contain-a-term-in-elasticsearch

2. AWS ES는 Domain이 생성된 이후에 인스턴스 type이나 수의 자유성

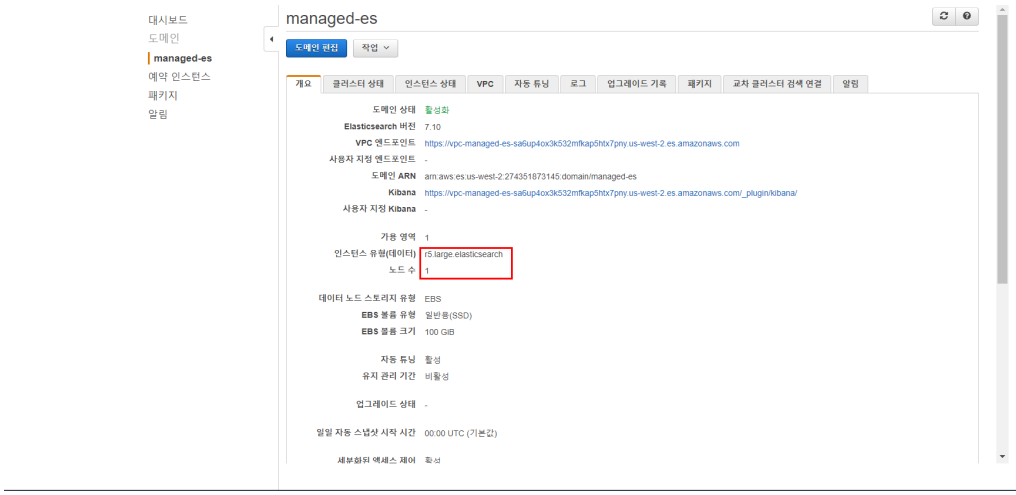
1. Elasticsearch 최초 :
    - 인스턴스 유형(데이터) : `r6.large.elasticsearch`
    - 노드 수 : `1`



2. 노드 수 변경 :
    - 인스턴스 유형(데이터) : `r6.large.elasticsearch`
    - 노드 수 : `2`



3. 인스턴스 유형 변경 :
    - 인스턴스 유형(데이터) : `r5.large.elasticsearch`
    - 노드 수 : `1`



참고 URL :
https://aws.amazon.com/ko/premiumsupport/knowledge-center/elasticsearch-scale-up/

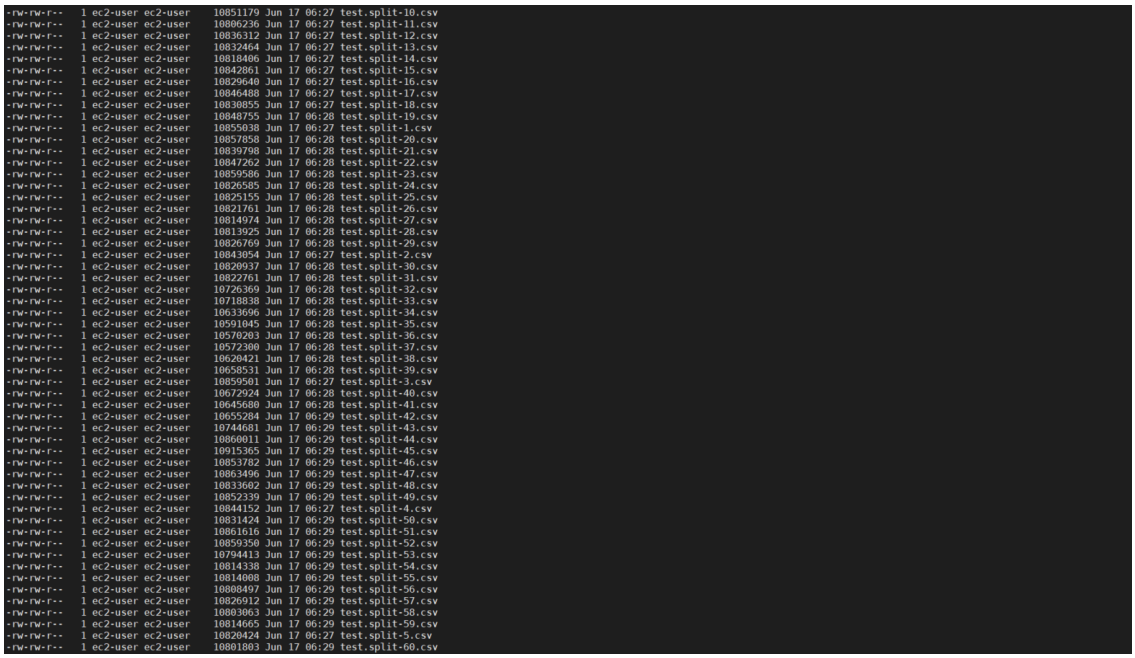
3. Data를 분할하여 Migration 가능 여부

1. Snapshot (shard 단위)
   아니요. 최초 snapshot은 index 단위로 해야합니다 하지만 2번 째부터는 증분 형태의 snapshot이므로 최초 시도보다 빠르게 migration 가능합니다

2. Elasticdump
   가능합니다 file size를 option으로 설정하면 해당 용량 만큼 partitioning 되어 적재 됩니다.

```
# 예시 script

/home/ec2-user/node_modules/elasticdump/bin/elasticdump\
  --input="http://elastic:Bespin12@10.0.1.49:9200/livechat"\
  --output="/home/ec2-user/test.csv"\
  --limit=10000\
  --fileSize=10mb
```

결과



4. 다른 Version의 ES 끼리 Data Migration 방법

1. 중간 호환 version 사용
2. Elasticdump 사용

1. 중간 호환 version 사용

참고 URL :
https://www.elastic.co/kr/support/matrix#matrix_compatibility

1. Source(낮은 ver), Target(높은 ver) 일 경우 호환 가능한 중간 version을 통해 migration 가능
   Source(낮은 ver) > 중간 version(Source와 Target 호환) > Target(높은 ver) (2번의 mig 작업)
2. Source(높은 ver), Target(낮은 ver) 일 경우
   1. 호환 가능한 중간 version을 통해 migration 가능
      Source(높은 ver) > 중간 version(Source와 Target 호환) > reindexing 작업 > Target(높은 ver) (3번의 작업)
   2. Raw Data 추출 및 재적재 작업
      Source에서 Data (CSV, JSON) 추출 (Kibana Export, Python Client)
      Target에서 Data 적재 (Python Client)

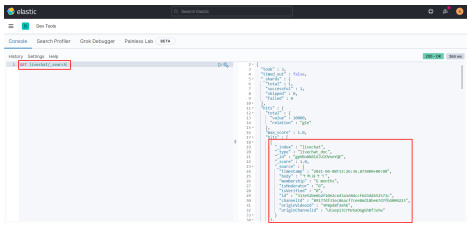
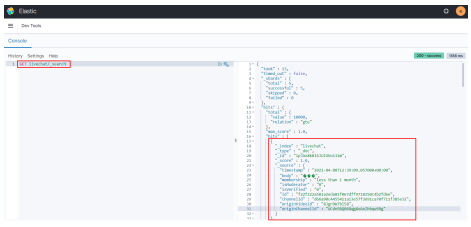
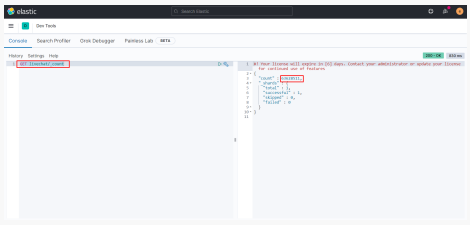
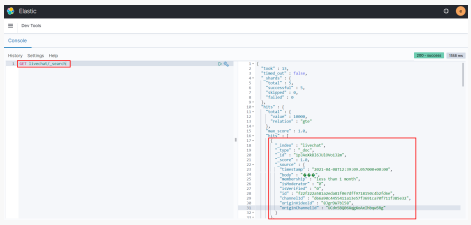
2. Elasticdump 사용

Elasticdump를 사용하면 Source 적재 되어 있는 Data를 조회하여 Target에 Insert 작업 가능합니다.
하지만 Elastico에서 인정한 공식 Migration 방법은 아닙니다

```
# aws elasticsearch http auth file
vi awsAuthFile.txt

# 하단 내용 추가
# ================================
user={username}
password={password}
```

```
# 예시

/home/ec2-user/node_modules/elasticdump/bin/elasticdump\
  --input="http://{username}:{password}@{Master_Private_IP}:9200/{Index_Name}"\
  --output="https://{AWS_Elasticsearch_DNS}/{Index_Nmae}"\
  --type=data\
  --limit=10000\
  --httpAuthFile=/home/ec2-user/awsAuthFile.txt
```

결과 : Source와 동일한 Document가 적재 됩니다.

유형	Source	Target
Content		
Count		

5. ES의 Data를 csv 혹은 Json으로 Export하는 방법

1. Kibana reporting 기능
2. Kibana DataTable
3. Elasticdump
4. Python client

1. Kibana reporting 기능

1. yml 파일 수정

> kibana.yml에서
> xpack.reporting.csv.maxSizeBytes : 100857600
> 상단 내용 추가
> 값은 바이트 단위입니다.
> 기본값은 10485760 (10MB) 입니다.
>
> 설정할 수 있는 최대 값은 100 MB 입니다.
>
> 참고 URL :
> https://discuss.elastic.co/t/reporting-csv-export-reached-the-max-size-and-contains-partial-data/116626/3

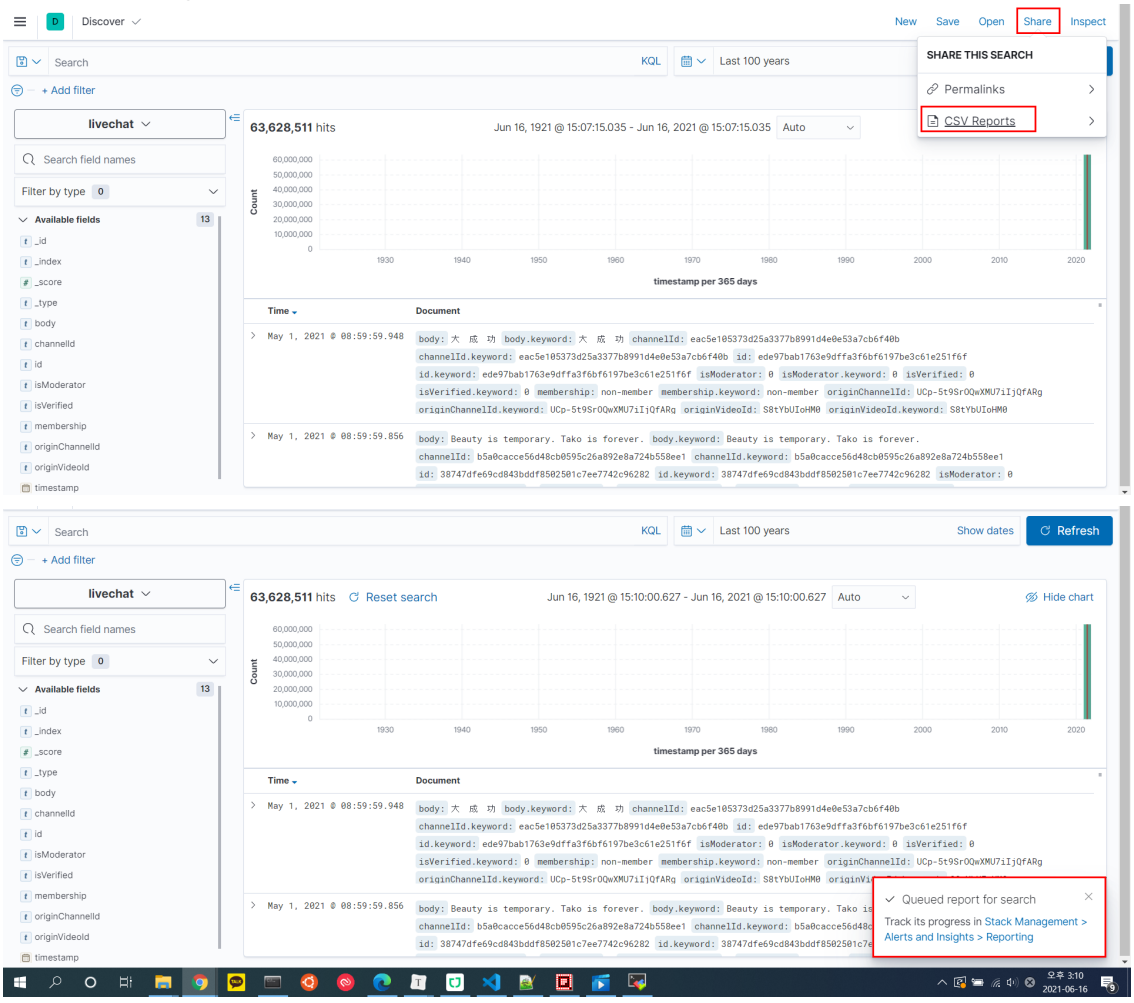
```
sudo vim /etc/kibana/kibana.yml
```

```
xpack.reporting.csv.maxSizeBytes : 100857600
```

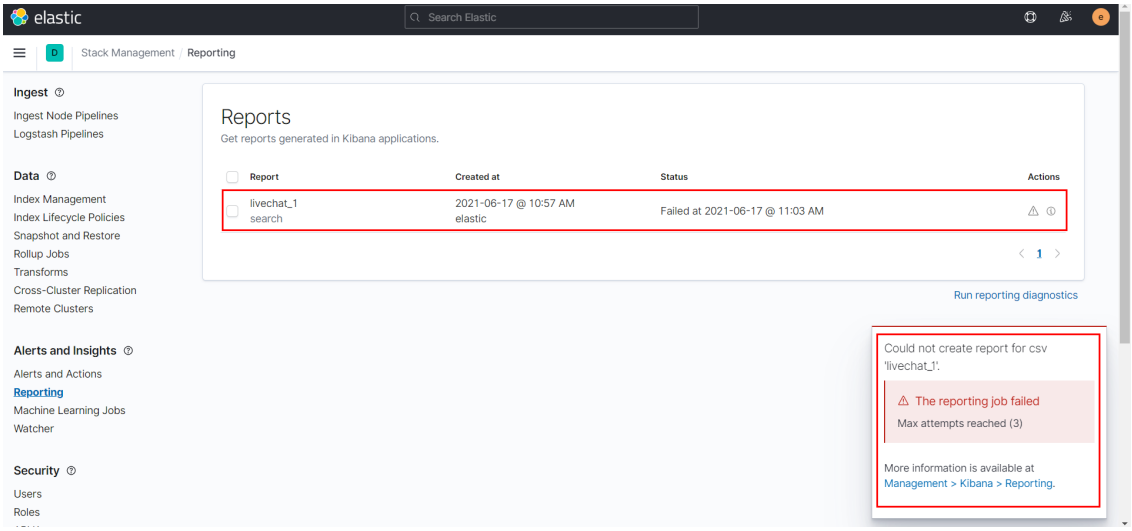
```
# List of Kibana client-side headers to send to Elasticsearch. To send *no* client-side
# headers, set this value to [] (an empty list).
#elasticsearch.requestHeadersWhitelist: [ authorization ]

# Header names and values that are sent to Elasticsearch. Any custom headers cannot be overwritten
# by client-side headers, regardless of the elasticsearch.requestHeadersWhitelist configuration.
#elasticsearch.customHeaders: {}

# Time in milliseconds for Elasticsearch to wait for responses from shards. Set to 0 to disable.
#elasticsearch.shardTimeout: 30000

# Logs queries sent to Elasticsearch. Requires logging.verbose set to true.
#elasticsearch.logQueries: false

# Specifies the path where Kibana creates the process ID file.
#pid.file: /run/kibana/kibana.pid

# Enables you to specify a file where Kibana stores log output.
logging.dest: /var/log/kibana.log

# Set the value of this setting to true to suppress all logging output.
#logging.silent: false

# Set the value of this setting to true to suppress all logging output other than error messages.
#logging.quiet: false

# Set the value of this setting to true to log all events, including system usage information
# and all requests.
#logging.verbose: false

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000

# Specifies locale to be used for all localizable strings, dates and number formats.
# Supported languages are the following: English - en , by default , Chinese - zh-CN .
#i18n.locale: "en"
#xpack.security.enabled

xpack.reporting.csv.maxSizeBytes: 100857600
"/etc/kibana/kibana.yml" 118L, 5320C                          117,1         Bot
```

2. Kibana Discover에서 해당 Index 전체 조회

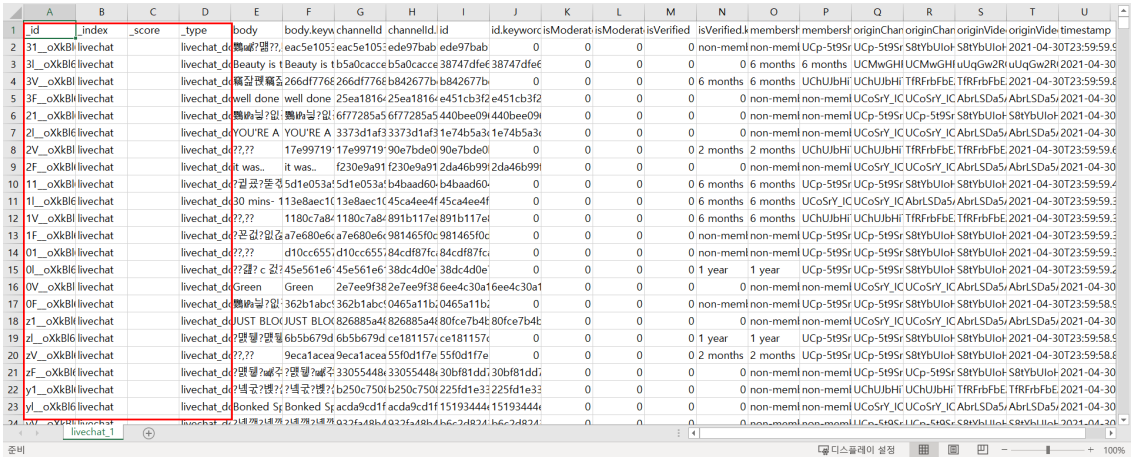
3. Share > CSV Reports



4. Stack Management > Alerts and Insights > Reporting
   해당 내용은 Export가 실패합니다.
   전체 Data 용량이 약 11GB 인데 수용 최대 용량이 100MB 이기 때문입니다.



해당 방법을 이용하기 위해서는 100MB 정도의 Data만 Export 해야합니다
불필요 Column이 포함 됩니다.

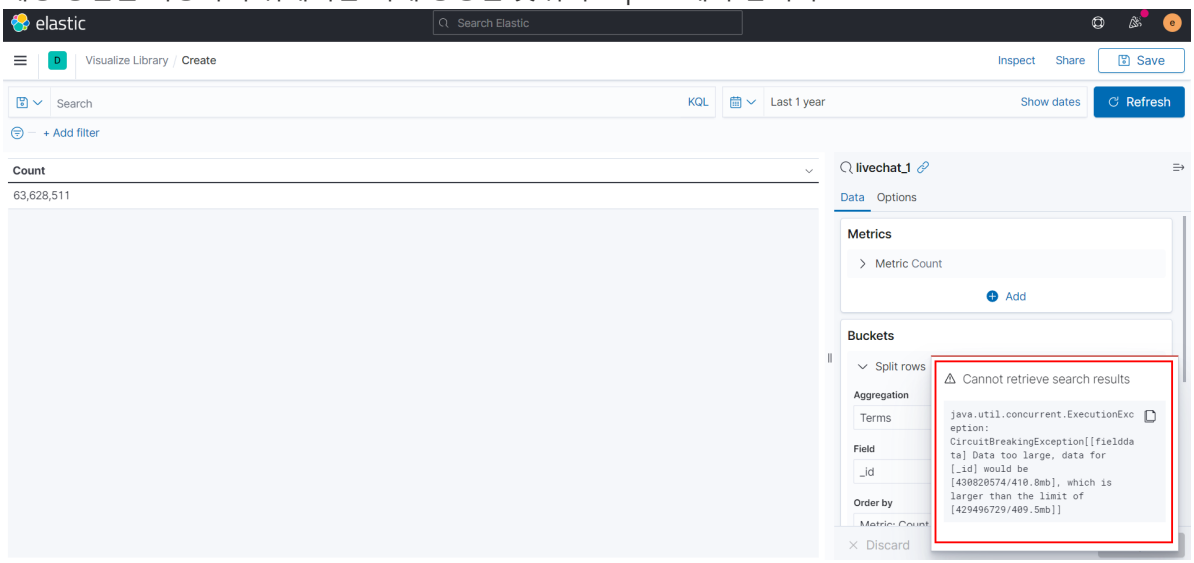


2. Kibana DataTable

1. Kibana > Visualize > Data Table
2. Metric : `Count`
3. Split rows
    1. Aggregation : `Terms`
    2. Field : `_id`

해당 내용도 수용 용량 초과로 실패합니다.

해당 방법을 이용하기 위해서는 최대 용량을 맞춰서 Export 해야 합니다.



3. Elasticdump

1. Node Js 설치

   AMAZON Linux 기준 설치

```
sudo yum update -y

# npm 설치
https://docs.aws.amazon.com/ko_kr/sdk-for-javascript/v2/developer-guide/setting-up-node-on-ec2-instance.html

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash

. ~/.nvm/nvm.sh

nvm install node

node -e "console.log('Running Node.js ' + process.version)"
```

2. Elasticdump 설치

```
npm install elasticdump

# Home 디렉토리에 Elasticdump 설치
```

3. Elasticsearch httpauth용 파일 작성

```
vi httpAuthFile.txt

# 하단 내용 추가
# ================================
user={username}
password={password}
```

4. Dump 작업용 Shell script 작성
   참고 URL :
   https://github.com/elasticsearch-dump/elasticsearch-dump

```
vi json_export.sh
```

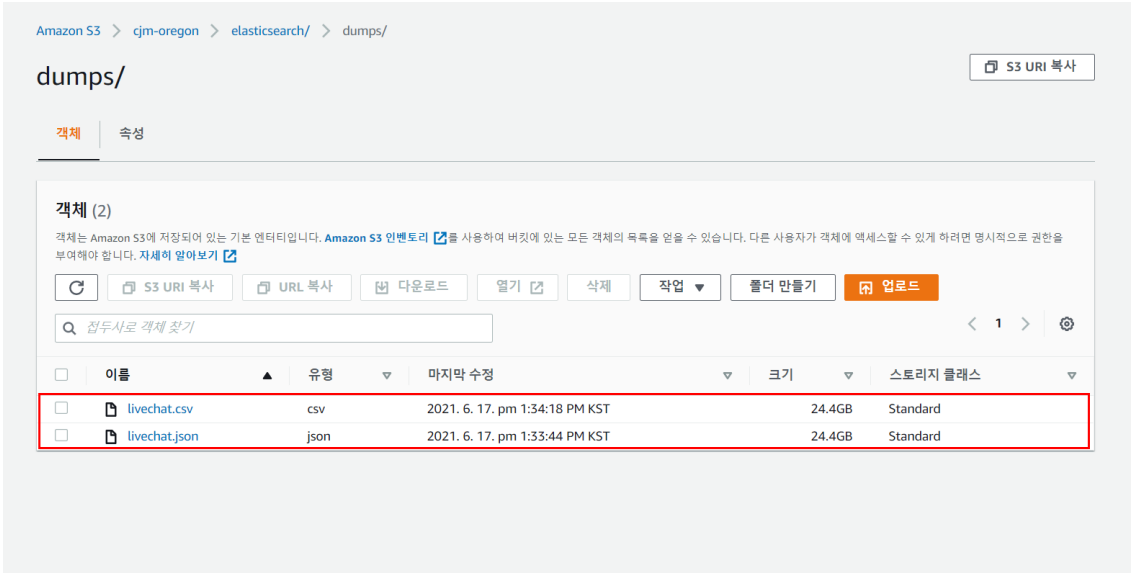
하단 내용 입력

```
#! /bin/sh

/home/ec2-user/node_modules/elasticdump/bin/elasticdump\
  --s3AccessKeyId "{ACCESS_KEY}" \
  --s3SecretAccessKey "{SECRET_KEY}" \
  --input=http://{Master_Private_IP}:9200/{Index_Name}\
  --output "s3://{Bucket_Name}/{Path}/{Target_File_Name}"\
  --limit=10000 \
  --httpAuthFile=/home/ec2-user/httpAuthFile.txt
```

5. Daemon으로 Shell Script 수행

```
nohup /home/ec2-user/json_export.sh > dump_json.log &
```

```
Thu, 17 Jun 2021 01:35:23 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:23 GMT | got 100 objects from source elasticsearch (offset: 500000)
Thu, 17 Jun 2021 01:35:23 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:28 GMT | got 100 objects from source elasticsearch (offset: 500100)
Thu, 17 Jun 2021 01:35:28 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:28 GMT | got 100 objects from source elasticsearch (offset: 500200)
Thu, 17 Jun 2021 01:35:28 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:28 GMT | got 100 objects from source elasticsearch (offset: 500300)
Thu, 17 Jun 2021 01:35:28 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:28 GMT | got 100 objects from source elasticsearch (offset: 500400)
Thu, 17 Jun 2021 01:35:28 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:28 GMT | got 100 objects from source elasticsearch (offset: 500500)
Thu, 17 Jun 2021 01:35:28 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:33 GMT | got 100 objects from source elasticsearch (offset: 500600)
Thu, 17 Jun 2021 01:35:33 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:33 GMT | got 100 objects from source elasticsearch (offset: 500700)
Thu, 17 Jun 2021 01:35:33 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:33 GMT | got 100 objects from source elasticsearch (offset: 500800)
Thu, 17 Jun 2021 01:35:33 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:33 GMT | got 100 objects from source elasticsearch (offset: 500900)
Thu, 17 Jun 2021 01:35:33 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:33 GMT | got 100 objects from source elasticsearch (offset: 501000)
Thu, 17 Jun 2021 01:35:33 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:38 GMT | got 100 objects from source elasticsearch (offset: 501100)
Thu, 17 Jun 2021 01:35:38 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:38 GMT | got 100 objects from source elasticsearch (offset: 501200)
Thu, 17 Jun 2021 01:35:38 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:38 GMT | got 100 objects from source elasticsearch (offset: 501300)
Thu, 17 Jun 2021 01:35:38 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:38 GMT | got 100 objects from source elasticsearch (offset: 501400)
Thu, 17 Jun 2021 01:35:38 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:38 GMT | got 100 objects from source elasticsearch (offset: 501500)
Thu, 17 Jun 2021 01:35:38 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:43 GMT | got 100 objects from source elasticsearch (offset: 501600)
Thu, 17 Jun 2021 01:35:43 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:43 GMT | got 100 objects from source elasticsearch (offset: 501700)
Thu, 17 Jun 2021 01:35:43 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:43 GMT | got 100 objects from source elasticsearch (offset: 501800)
Thu, 17 Jun 2021 01:35:43 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:43 GMT | got 100 objects from source elasticsearch (offset: 501900)
Thu, 17 Jun 2021 01:35:43 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:43 GMT | got 100 objects from source elasticsearch (offset: 502000)
Thu, 17 Jun 2021 01:35:43 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:48 GMT | got 100 objects from source elasticsearch (offset: 502100)
Thu, 17 Jun 2021 01:35:48 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:48 GMT | got 100 objects from source elasticsearch (offset: 502200)
Thu, 17 Jun 2021 01:35:48 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:48 GMT | got 100 objects from source elasticsearch (offset: 502300)
Thu, 17 Jun 2021 01:35:48 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:48 GMT | got 100 objects from source elasticsearch (offset: 502400)
Thu, 17 Jun 2021 01:35:48 GMT | sent 100 objects to destination s3, wrote 100
Thu, 17 Jun 2021 01:35:48 GMT | got 100 objects from source elasticsearch (offset: 502500)
Thu, 17 Jun 2021 01:35:48 GMT | sent 100 objects to destination s3, wrote 100
^C
```



4. Python client

1. python Script 작성

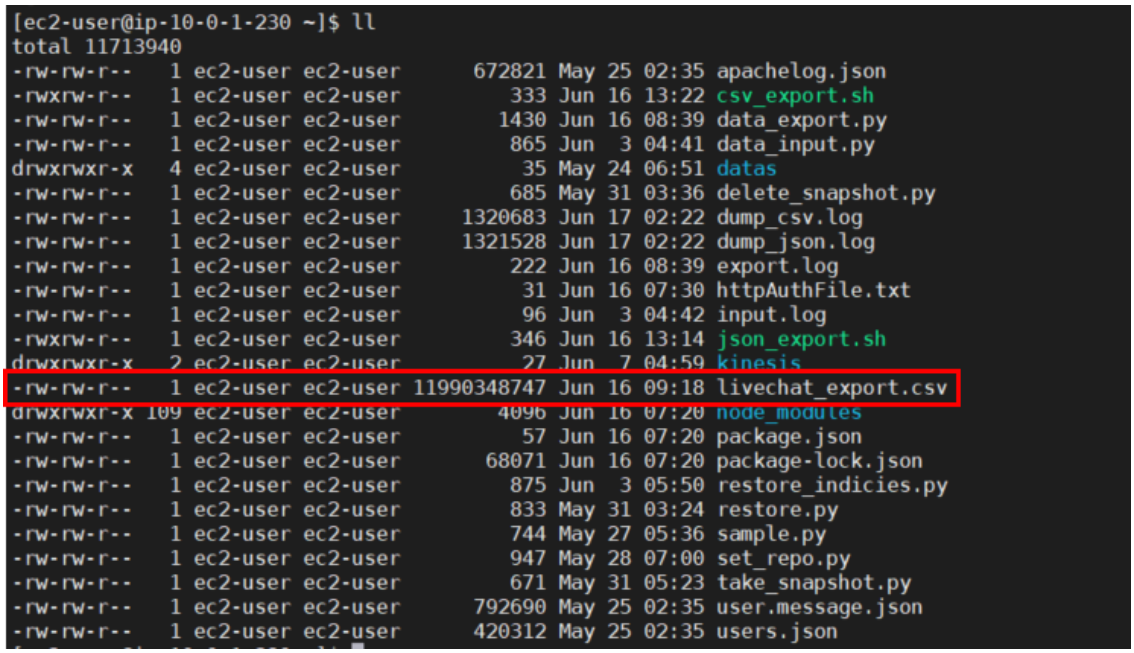
```
vi data_export.py
```

```
from elasticsearch import Elasticsearch
from pprint import pprint
import csv
import os

# Column list를 뽑기 위한 함수
def search_api(es, index_name):
    index = index_name
    body = {
        'size':1,
        'query':{
            'match_all':{}
        }
    }
    res = es.search(index=index, body=body)
    return res

es = Elasticsearch(
    # 마스터 계정정보를 이용하여 http로 ES와 통신
    hosts = [{'host': '{host}', 'port': '{port}'}],
    http_auth = ('{username}', '{password}'),
    scheme="http"
)

indices = [""]
csv_columns = list(search_api(es, indices[0])['hits']['hits'][0]
['_source'].keys())
csv_file = os.environ['HOME']+'/'+indices[0]+'_export.csv'

# Elasticsearch의 _search API는 Return 최대 Size가 10000 입니다
# 따라서 Scroll_id로 계속 Scroll 하며 추출해야 합니다.
for index in indices:
    result = es.search(
        index =index,
        doc_type =index+"_doc",
        scroll ='5m',
        body = {
            'size':10000,
            'query':{
                'match_all':{}
            }
        }
    )
    scroll_id = result['_scroll_id']
    scroll_size = result['hits']['total']['value']
    while scroll_size > 0:
        with open(csv_file, 'a') as csvfile:
            writer = csv.DictWriter(csvfile, fieldnames=csv_columns)
```

```
            writer.writeheader()
            for doc in result['hits']['hits']:
                writer.writerow(doc['_source'])

        csvfile.close()

        result = es.scroll(scroll_id = scroll_id, scroll ='5m')

        scroll_id = result['_scroll_id']
        scroll_size = len(result['hits']['hits'])
```

2. Python Script daemon 실행

```
nohup python3 -u data_export.py > export.log &
```



참고 URL :
https://hevodata.com/learn/elasticsearch-export/#b2
https://github.com/higee/elastic/issues/20