使用 Elasticdump 進行資料轉移

處理對象

舊機	PROD-ElasticSearch-infra-01	10.23.1.167
舊機	PROD-ElasticSearch-infra-02	10.23.1.168
舊機	PROD-ElasticSearch-infra-03	10.23.1.169
新機	PROD-ElasticSearch-infra-01	10.23.1.16
新機	PROD-ElasticSearch-infra-02	10.23.1.193
新機	PROD-ElasticSearch-infra-03	10.23.1.96

安裝 elasticdump 軟體

- 在新的 Server 執行

https://blog.csdn.net/Ache_csdn/article/details/116029509

工程師的江湖

操作目的：將 ELK-01 數據遷移至 ELK-02 （單機遷移） sudo yum install epel-release 啟用EPEL存儲庫後，運行以下命令以添加Node.js v6 LTS存儲庫： curl --silent --location https://rpm.nodesource.com/setup_6.x | sudo bash - 啟用NodeSource存儲庫後，我們可以繼續執行Node.js v6 LTS和npm安裝： sudo yum install

https://dotblogs.com.tw/xerion30476/2020/07/04/182915

```
# 查看所有 index
curl http://localhost:9200/_cat/indices
curl -XGET 'http://localhost:9200/_cat/indices'

# 查看所有 index，有 header
curl http://localhost:9200/_cat/indices?v
curl -XGET 'http://localhost:9200/_cat/indices?v'

# 查看特定 index
curl http://localhost:9200/_cat/indices/service-logs_10
curl -XGET 'http://localhost:9200/_cat/indices/service-logs_10'

# 查看特定 index，有 header
curl http://localhost:9200/_cat/indices/service-logs_10?v
curl -XGET 'http://localhost:9200/_cat/indices/service-logs_10?v'

# 查看特定 index，模糊查詢
curl http://localhost:9200/_cat/indices/service-logs*
curl -XGET 'http://localhost:9200/_cat/indices/service-logs*'

# 查看特定 index，模糊查詢有 header
curl http://localhost:9200/_cat/indices/service-logs*?v
curl -XGET 'http://localhost:9200/_cat/indices/service-logs*?v'



# 查看 cluster 中的 nodes 資訊
curl 'localhost:9200/_cat/nodes?v'

# 查看 cluster 中的 shards 資訊
curl -s "http://localhost:9200/_cat/shards?v"

# 查看 cluster 中的 health 資訊
curl -XGET 'localhost:9200/_cluster/health?pretty'

# 確認 service 的 process 是否正在執行
ps -aux | grep elasticdump.sh
```

修改 mapping 內容

- 說明：dump 出來的 mapping 格式與原本在 Kibana 所看到的格式有差異，所以會導致錯誤；取出後需要先修正才能再匯入。

原始版本	dump 後版本
{ "mappings": { "*doc": { "dynamic_templates": [{ "internal_fields": { "mapping": { "type": "keyword" }, "match": "gl2*", "match_mapping_type": "string" } }, { "store_generic": { "mapping": { "type": "keyword" }, "match_mapping_type": "string" } }], "properties": { "@metadata": { "type": "keyword" }, "@timestamp": { "type": "date" }, "@version": { "type": "keyword" }, "agent": { "type": "keyword" }, "args": { "type": "keyword" }, "body_bytes_sent": { "type": "long" }, "cloud": { "type": "keyword" }, "ecs": { "type": "keyword" }, "fields": { "type": "keyword" }, "forwarded_for": { "type": "keyword" }, "forwarded_for_city_name": { "type": "keyword" }, "forwarded_for_country_code": { "type": "keyword" }, "forwarded_for_geolocation": { "type": "keyword" }, "full_message": { "analyzer": "standard", "type": "text" }, "gl2_accounted_message_size": { "type": "long" }, "gl2_message_id": { "type": "keyword" }, "gl2_processing_timestamp": { "format": "uuuu-MM-dd HH:mm:ss.SSS", "type": "date" }, "gl2_receive_timestamp": { "format": "uuuu-MM-dd HH:mm:ss.SSS", "type": "date" }, "gl2_source_input": { "type": "keyword" }, "gl2_source_node": { "type": "keyword" }, "host": { "type": "keyword" }, "hostname": { "type": "keyword" }, "hosts": { "type": "keyword" }, "hosts_city_name": { "type": "keyword" }, "hosts_country_code": { "type": "keyword" }, "hosts_geolocation": { "type": "keyword" }, "http_referrer": { "type": "keyword" }, "http_user_agent": { "type": "keyword" }, "input": { "type": "keyword" }, "log": { "type": "keyword" }, "message": {	{ "service-logs_11": { "mappings": { "dynamic_templates": [{ "internal_fields": { "match": "gl2_*", "match_mapping_type": "string", "mapping": { "type": "keyword" } } }, { "store_generic": { "match_mapping_type": "string", "mapping": { "type": "keyword" } } }], "properties": { "@metadata": { "type": "keyword" }, "@timestamp": { "type": "date" }, "Kafka_topic": { "type": "keyword" }, "action": { "type": "keyword" }, "agent": { "type": "keyword" }, "channel": { "type": "keyword" }, "class": { "type": "keyword" }, "cloud": { "type": "keyword" }, "ecs": { "type": "keyword" }, "fields": { "type": "keyword" }, "full_message": { "type": "text", "analyzer": "standard" }, "gl2_accounted_message_size": { "type": "long" }, "gl2_message_id": { "type": "keyword" }, "gl2_processing_error": { "type": "keyword" }, "gl2_processing_timestamp": { "type": "date", "format": "uuuu-MM-dd HH:mm:ss.SSS" }, "gl2_receive_timestamp": { "type": "date", "format": "uuuu-MM-dd HH:mm:ss.SSS" }, "gl2_source_input": { "type": "keyword" }, "gl2_source_node": { "type": "keyword" }, "host": { "type": "keyword" }, "input": { "type": "keyword" }, "level": { "type": "keyword" }, "line": { "type": "keyword" }, "log": { "type": "keyword" }, "message": { "type": "text", "analyzer": "standard" }, "service": { "type": "keyword" }, "source": { "type": "text", "analyzer": "analyzer_keyword", "fielddata": true }, "source_ip": { "type": "keyword" }, "span": { "type": "keyword" }, "streams": { "type": "keyword" }, "thread": { "type": "keyword" }, "timestamp": { "type": "date", "format": "uuuu-MM-dd HH:mm:ss.SSS" }, "trace": { "type": "keyword" }, "user": { "type": "keyword" } } } } }

"analyzer": "standard", "type": "text" }, "my_host": { "type": "keyword" },
"my_host_city_name": { "type": "keyword" },
"my_host_country_code": { "type": "keyword" },
"my_host_geolocation": { "type": "keyword" }, "real_ip": { "type": "keyword" },
"real_ip_city_name": { "type": "keyword" },
"real_ip_country_code": { "type": "keyword" },
"real_ip_geolocation": { "type": "keyword" }, "remote_addr": { "type": "keyword" },
"remote_addr_city_name": { "type": "keyword" },
"remote_addr_country_code": { "type": "keyword" },
"remote_addr_geolocation": { "type": "keyword" },
"remote_port": { "type": "long" },
"request": { "type": "keyword" },
"request_body": { "type": "keyword" }, "request_method": { "type": "keyword" },
"request_time": { "type": "float" },
"request_url": { "type": "keyword" }, "scheme": { "type": "keyword" }, "server_addr": { "type": "keyword" },
"server_addr_city_name": { "type": "keyword" },
"server_addr_country_code": { "type": "keyword" },
"server_addr_geolocation": { "type": "keyword" },
"server_name": { "type": "keyword" }, "server_port": { "type": "long" },
"server_protocol": { "type": "keyword" }, "service": { "type": "keyword" }, "source": { "analyzer": "analyzer_keyword", "fielddata": true, "type": "text" },
"status": { "type": "keyword" },
"streams": { "type": "keyword" },
"tags": { "type": "keyword" },
"timestamp": { "format": "uuuu-MM-dd HH:mm:ss.SSS", "type": "date" }, "upstream_addr": { "type": "keyword" },
"upstream_cache_status": { "type": "keyword" }}}}} },
"upstream_response_time": { "type": "float" }, "uri": { "type": "keyword" } } } } }

主程式

```
#!/bin/bash
#####################################
# Project: Elasticdump Implementation
# Branch:
# Author: Gok, the DBA
# Created: 2022-09-20
# Updated: 2022-10-25
# Note:
#####################################

start_at=`date`
source_ip='10.23.1.168'
local_ip=$(hostname -i)
es_command=`sudo find / -name "elasticdump" -type f`
dest_dir='/data/es'
mkdir -p /data/es

## Install the jq for json readibility, execute if not ready
# sudo yum install epel-release -y
# sudo yum update -y
# sudo yum install jq -y
# rpm -qa | grep -i jq
jq --version

title='Execute elasticdump Info'
token="5624325337:AAEAhFxz8FitLOE6ez3FyErRaRXlfLOsPEc"
chat="-675619128"

#mapping='service-logs nginx-logs'
mapping='service-logs'
for idx in $mapping;
do
    i=`curl -XGET "http://${source_ip}:9200/_cat/indices/${idx}*?&s=creation.date:desc,docs.count" | grep -v 'close' | awk '{print $3}' |

    echo ''
    echo currently is working on $i

    #### Note ########################################################
    # limit：傳輸資料筆數，最大 30000，default = 100；                    #
    #        上傳 elasticsearch 會等太久而 connection_timeout，故設定為 25000 #
    # concurrencyInterval：request 間隔，default = 5000 毫秒(ms)          #
    ##################################################################

    echo ''
    echo "Export Index $i"
    sudo $es_command --input=http://${source_ip}:9200/$i --output=$dest_dir/${i}.index --all=true --type=mapping --limit=30000 --concurrenc
    jq '.[]' $dest_dir/${i}.index | sudo tee $dest_dir/${i}_pre.index
    sudo sed -i 's/"mappings": {/"mappings": {"_doc": {/' $dest_dir/${i}_pre.index
    echo '}' | sudo tee -a $dest_dir/${i}_pre.index
    sudo mv $dest_dir/${i}_pre.index $dest_dir/${i}.index


    echo ''
    echo "Export Data $i"
    sudo $es_command --input=http://${source_ip}:9200/$i --output=$dest_dir/${i}.json --all=true --type=data --limit=30000 --concurrencyInt


    echo ''
    echo "Import Index $i"
    sudo $es_command --input=$dest_dir/${i}.index --output=http://localhost:9200 --bulk=true --type=mapping --limit=30000 --concurrencyInte

    echo ''
    echo "Import data $i"
    sudo $es_command --input=$dest_dir/${i}.json --output=http://localhost:9200 --bulk=true --type=data --limit=25000 --concurrencyInterval


    msg="the index $i is finished between $start_at at `date`"
    curl -X POST "https://api.telegram.org/bot${token}/sendMessage" -d "chat_id=${chat}&text=${title} at [[`hostname`]]

${msg}" > /dev/null 2>&1
done
```

遭遇錯誤 es_rejected_execution_exception

- 解法：**reduce the request size** (官方不建議) or **allocate more memory to ES**
- 參考來源

加大可用記憶體給 ElasticSearch

```
$ vim /etc/elasticsearch/jvm.options

## 把以下兩個參數調大就對了
## -Xms4g
## -Xmx4g
-Xms10g
-Xmx10g
```

遭遇錯誤 TOO_MANY_REQUESTS 或其他

簡 https://www.jianshu.com/p/b0fa9e698c8a

解方

- 以指令 `df -h` 查看 server 的硬碟容量使用狀況，若佔用太多則請盡量清空，保持可用空間相對較多，則許多錯誤不會再發生。

官方文件

https://github.com/elasticsearch-dump/elasticsearch-dump