

# 台灣科技大學 社群媒體分析實務 第一次作業

電機工程系 M10507514 蘇振驊

# 第一題

建構個人資料分析平台之說明

作業系統 : *Mac OS Sierra*

硬體資源 : *Macbook pro*

*Jupyter, Elasticsearch, Logstash*

安裝 :

## Lab 2: Analytics Platform Construction

- Please download the Docker YML  
file : [https://dl.dropboxusercontent.com/u/23229197/NTU\\_course2016/yml/jupyter%20elasticsearch%20logstash.zip](https://dl.dropboxusercontent.com/u/23229197/NTU_course2016/yml/jupyter%20elasticsearch%20logstash.zip)
- Unzipping: docker-compose.yml
- execute command : docker-compose up

## Run Jupyter, Elasticsearch, Logstash in Docker Engine :

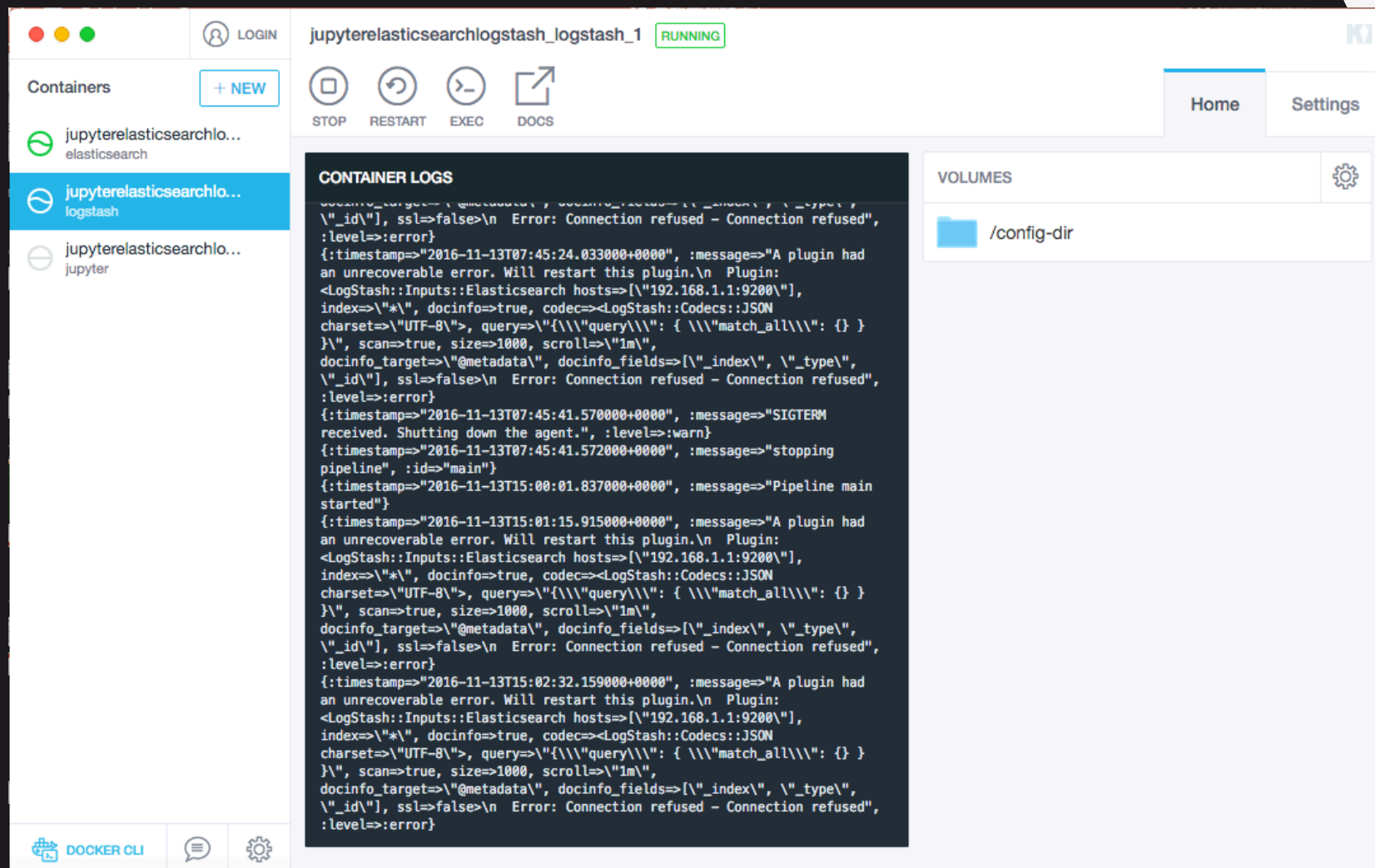
- 使用Lab2連結，下載完成後，進去directory, 然後execute command :  
`docker compose -up.`
- 系統就會自己下載docker, jupyter, elasticsearch, logstash

```
jupyter+elasticsearch+logstash — docker-compose • docker-compose up — 8...
[chsude-MBP:jupyter+elasticsearch+logstash chsu$ ls
README.md                               example-for-scikitlearn.py
docker-compose.yml                      example-for-tensorflow.py
example-for-elasticsearch.py            logstash.config
[chsude-MBP:jupyter+elasticsearch+logstash chsu$ docker-compose up
Pulling es-server (lucasko/elasticsearch:latest)...
latest: Pulling from lucasko/elasticsearch
5c90d4a2d1a8: Downloading [=====>
44.04 MB/51.35 MBload complete
c6072700a242: Download complete
42.19 MB/42.53 MBload complete
620b5227cf38: Download complete
3cfd33220efa: Download complete
864a98a84dd2: Downloading [=====>
23.77 MB/130 MBload complete
284.4 kB/284.4 kBting
fad43f822918: Waiting
d4cd106ced0d: Waiting
882de15fab49: Waiting
e4b13e908063: Waiting
8d1aacf525cb: Waiting
ef95375f815b: Waiting
f495e9d95d28: Waiting
```

## 額外安裝其它套件

套件名稱：Kitematic

用途：方便Docker管理  
container



## 第二題

- a小題：共計有多少Twitter參與這類關鍵字議題討論？ 關鍵字：C V E

```
In [3]: import pyes

conn = pyes.es.ES(server=[('http', 'localhost', 9200)])
bq = pyes.query.BoolQuery()
# BoolQuery本身是一個Query的組合，可以使用add_must(), add_must_not(), add_should()來使用。

bq.add_must(pyes.query.TermQuery(field="text", value="cve")) #(field, term)
#bq.add_must_not(pyes.query.TermQuery("name", "john")) #(field, term)

result = conn.search(query=bq, indices='twitter2' , doc_types='tweet')
#使用Boolquery來當query的值。

print "Tweets contain 'cve': ",len(result)  #tweets contain "cve"
StoreExceptDuplicats = set()
for x in result:
    StoreExceptDuplicats.add(x['uid'])

print "Twitter 總數:",len(StoreExceptDuplicats)
```

```
Tweets contain 'cve': 644
Twitter 總數: 71
```

## 關鍵字：Obama

```
In [10]: import pyes

conn = pyes.es.ES(server=[('http', 'localhost', 9200)])
bq = pyes.query.BoolQuery()
# BoolQuery本身是一個Query的組合，可以使用add_must(), add_must_not(), add_should()來使用。

bq.add_must(pyes.query.TermQuery(field="text", value="obama")) #(field, term)
#bq.add_must_not(pyes.query.TermQuery("name", "john")) #(field, term)

result = conn.search(query=bq, indices='twitter2', doc_types='tweet')
#使用Boolquery來當query的值。

print "Tweets contain 'obama': ", len(result) #tweets contain obama
StoreExceptDuplicats = set()
for x in result:
    StoreExceptDuplicats.add(x['uid'])

print "Twitter 總數: ", len(StoreExceptDuplicats)
```

Tweets contain 'obama': 1010

Twitter 總數: 92

## 關鍵字：vulnerability

```
In [12]: import pyes

conn = pyes.es.ES(server=[('http', 'localhost', 9200)])
bq = pyes.query.BoolQuery()
# BoolQuery本身是一個Query的組合，可以使用add_must(), add_must_not(), add_should()來使用。

bq.add_must(pyes.query.TermQuery(field="text", value="vulnerability")) #(field, term)
#bq.add_must_not(pyes.query.TermQuery("name", "john")) #(field, term)

result = conn.search(query=bq, indices='twitter2', doc_types='tweet')
#使用Boolquery來當query的值。

print "Tweets contain 'vulnerability': ", len(result) #tweets contain Vulnerability
StoreExceptDuplicats = set()
for x in result:
    StoreExceptDuplicats.add(x['uid'])

print "Twitter 總數: ", len(StoreExceptDuplicats)

Tweets contain 'Vulnerability': 2242
Twitter 總數: 96
```

## 第二題

- b小題：共計有多少Tweets？

```
In [11]: import pyes
#from elasticsearch import Elasticsearch
#es = Elasticsearch([{'host': 'localhost', 'port': 9200}])
#es_address='127.0.0.1:9200'
#conn = pyes.es.ES(es_address)
conn = pyes.es.ES(server=[('http', 'localhost', 9200)])

q = pyes.query.MatchAllQuery()

result = conn.search(query=q, indices='twitter2', doc_types='tweet')# 全部的tweets
print("全部的tweets : ") ,
print(len(result))

len(result)
```

全部的tweets : 392792

Out[11]: 392792









