# Celery task fail to write to Elasticsearch

# Issue

Celery task failed with these kinds of traceback (ES client read timeout):

Traceback (most recent call last):

File "/usr/local/lib/python3.8/site-packages/celery/app/trace.py", line 412, in trace\_task

R = retval = fun(\*args, \*\*kwargs)

File "/usr/local/lib/python3.8/site-packages/celery/app/trace.py", line 704, in \_\_protected\_call\_\_

return self.run(\*args, \*\*kwargs)

File "/usr/src/app/data-quality/task/lib.py", line 336, in wrapper

raise e

File "/usr/src/app/data-quality/task/lib.py", line 328, in wrapper

do\_func = func(\*args, \*\*kwargs)

File "/usr/src/app/data-quality/task/metrics/tasks.py", line 294, in data\_collection

raise e

File "/usr/src/app/data-quality/task/metrics/tasks.py", line 260, in data\_collection

merge\_result, rule\_exceptions = collect\_metrics\_with\_rule\_params(

File "/usr/src/app/data-quality/core/utils/collection.py", line 270, in collect\_metrics\_with\_rule\_params

res = run\_database(period, cur\_begin\_dt, cur\_end\_dt, metric\_rule, load\_common\_conf)

File "/usr/src/app/data-quality/core/utils/collection.py", line 27, in run\_database

return class\_obj(common\_conf, rule, period, begin\_dt, end\_dt).run()

File "/usr/src/app/data-quality/core/database/ch\_sql.py", line 113, in run

self.dump\_results\_to\_es(index\_name, data)

File "/usr/src/app/data-quality/core/database/base.py", line 78, in dump\_results\_to\_es

es\_clt.bulk\_put(one, index\_name)

File "/usr/src/app/data-quality/core/connector/elasticsearch\_req.py", line 133, in bulk\_put

self.bulk\_flush()

File "/usr/src/app/data-quality/core/connector/elasticsearch\_req.py", line 137, in bulk\_flush

self.es\_clt\_write.bulk(self.bulk\_lst)

File "/usr/local/lib/python3.8/site-packages/elasticsearch/client/utils.py", line 152, in \_wrapped

return func(\*args, params=params, headers=headers, \*\*kwargs)

File "/usr/local/lib/python3.8/site-packages/elasticsearch/client/\_\_init\_\_.py", line 455, in bulk

return self.transport.perform\_request(

File "/usr/local/lib/python3.8/site-packages/elasticsearch/transport.py", line 392, in perform\_request

raise e

File "/usr/local/lib/python3.8/site-packages/elasticsearch/transport.py", line 358, in perform\_request

status, headers\_response, data = connection.perform\_request(

File "/usr/local/lib/python3.8/site-packages/elasticsearch/connection/http\_requests.py", line 176, in perform\_request

raise ConnectionTimeout("TIMEOUT", str(e), e)

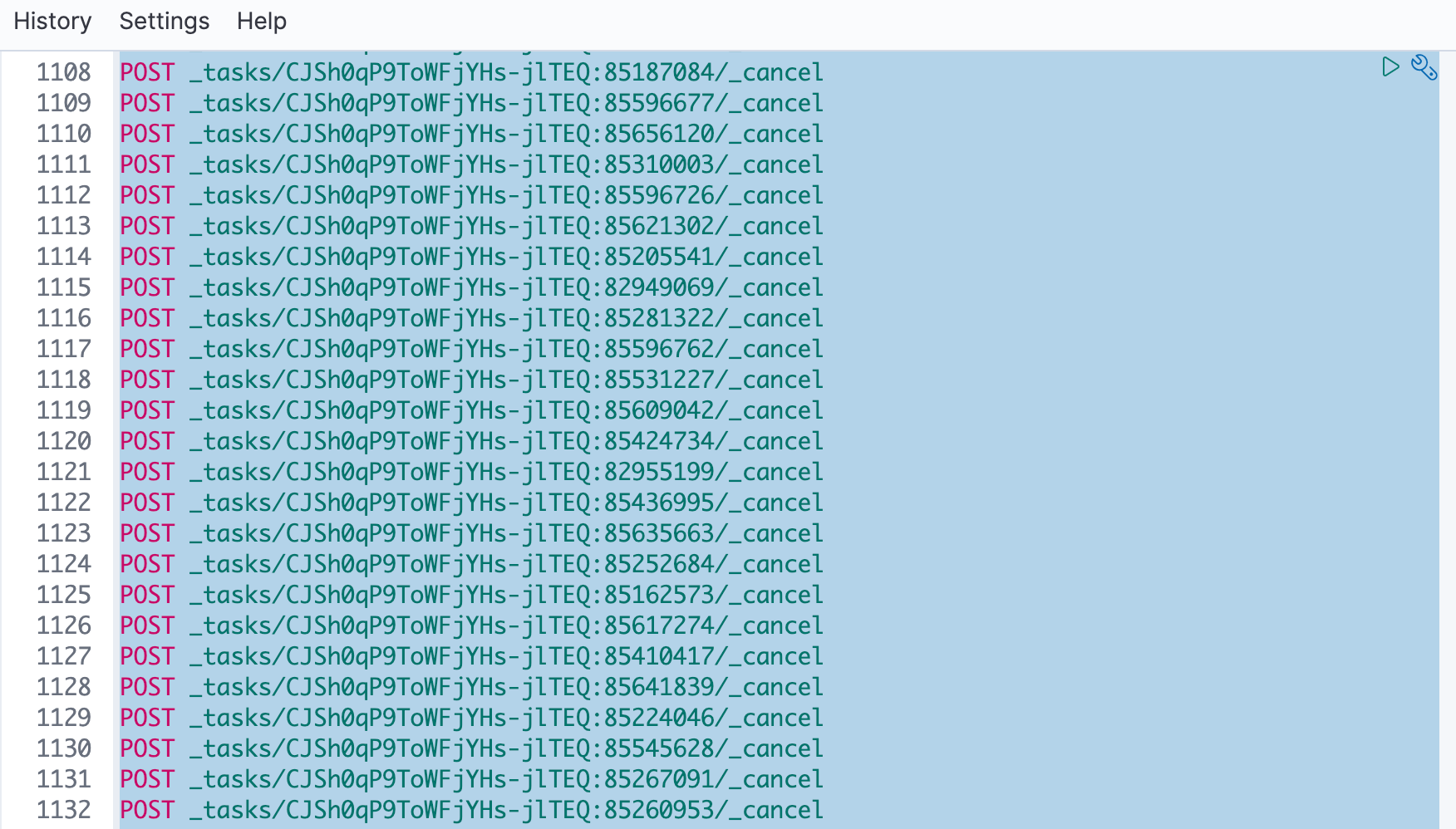
elasticsearch.exceptions.ConnectionTimeout: ConnectionTimeout caused by - ReadTimeout(HTTPSConnectionPool(host='esadsquality.vip.ebay.com', port=443): Read timed out. (read timeout=10))

# SOP

1. Check ES disk usage and CPU usage in <https://console.sherlock.io/d/vEO-N6qmz/production-monitoring?orgId=1&var-escluster=esadsquality-slc&var-name=All&from=now-3h&to=now>
2. If CPU usage is high for some ES instances (e.g. near 100%), check tasks using GET \_tasks in dev console of Kibana: <https://esadsquality-kibanaslc.vip.ebay.com/app/dev_tools#/console>
3. Find out the tasks which causing CPU usage high (usually there will be a lot of similar tasks). For example, if someone triggered some delete\_by\_query operations, we can get all related tasks using GET \_tasks?detailed=true&actions=\*/delete/byquery in dev console
4. Dump the tasks results to local json file, for example: [tasks.json](file:////download/attachments/1159673331/tasks.json%3fversion=1&modificationDate=1675504436000&api=v2)
5. Use the Python script to generate commands to cancel these delete\_by\_query operations (don't forget to change the file path and conditions in the code):
6. import json
7. with open('/Users/chonyu/Downloads/tasks.json') as f:
8. tasks = json.load(f)
9. for node in tasks['nodes']:
10. task\_list = tasks['nodes'][node]['tasks']
11. for task\_id in task\_list:
12. # You should replace the condition for your task
13. if 'delete-by-query [collect\_rule\_ch\_sre\_sales\_hourly\_business\_none\_replace]' == task\_list[task\_id]['description']:

print(f'POST \_tasks/{task\_id}/\_cancel')

1. Execute the commands in dev console (you can select multiple lines to execute them together), like:



1. Check again the CPU usage, the usage should decrease in a few minutes. And rerun the failed tasks, they should success.