Generate alerts using Elastalert

Step-by-step guide

Step 1:- Installation

Windows 10:-

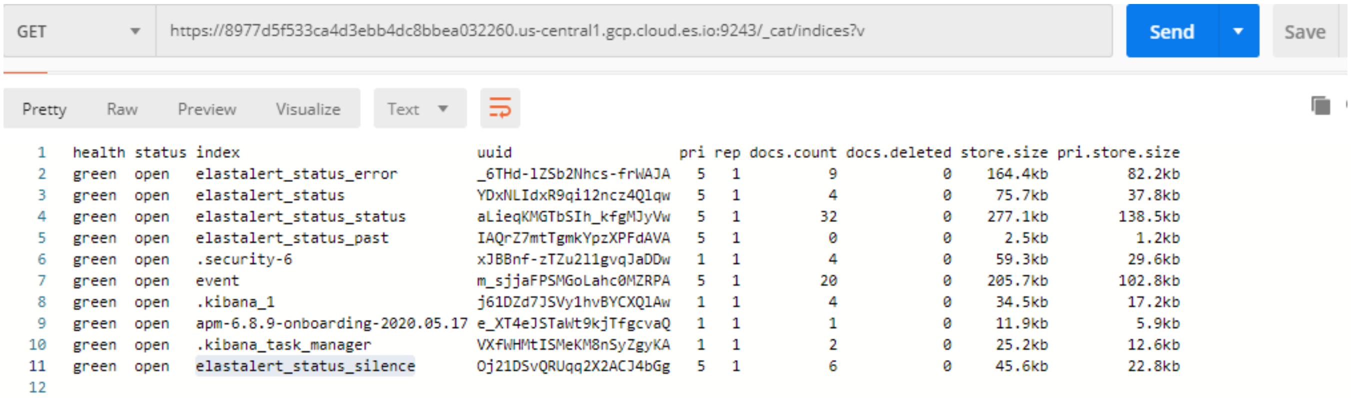
- Setup Elasticsearch Instance --> I used version 6.8.2 the instance is hosted in https://www.elastic.co/cloud/
- Install Python 3.6 -----> Install python 3.6.x from https://www.python.org/downloads/windows/
- Install ---> Microsoft Visual Build Tools
- Download Elastalert ----> git clone https://github.com/Yelp/elastalert.git
- From the Elastalert folder --> pip install -r requirements.txt
- From the Elastalert folder --> python setup.py install

For Linux:- https://elastalert.readthedocs.io/en/latest/running_elastalert.html

Note:- We can also run Elastalert on a container

Step 2:- Create Elastalert indexes in Elasticsearch to store alerts and errors

- From Elastalert Folder use this command to create required meta indexes --> elastalert-create-index
- This will create following indexes try list indexes from the cluster using https://host:port/_cat/indices?v



https://8977d5f533ca4d3ebb4dc8bbea032260.us-central1.gcp.cloud.es.io:9243/_cat/indices?v

```
health status index                          uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   elastalert_status_error        _6THd-1ZSb2Nhcs-frWAJA   5   1          9            0    164.4kb         82.2kb
green  open   elastalert_status              YDxNLIdxR9qi12ncz4Q1qw   5   1          4            0     75.7kb         37.8kb
green  open   elastalert_status_status       aLieqKMGTbSIh_kfgMJyVw   5   1         32            0    277.1kb        138.5kb
green  open   elastalert_status_past         IAQrZ7mtTgmkYpzXPFdAVA   5   1          0            0      2.5kb          1.2kb
green  open   .security-6                    xJBBnf-zTZu21lgvqJaDDw   1   1          4            0     59.3kb         29.6kb
green  open   event                          m_sjjaFPSMGoLahc0MZRPA   5   1         20            0    205.7kb        102.8kb
green  open   .kibana_1                      j61DZd7JSVy1hvBYCXQ1Aw   1   1          4            0     34.5kb         17.2kb
green  open   apm-6.8.9-onboarding-2020.05.17 e_XT4eJSTaWt9kjTfgcvaQ   1   1          1            0     11.9kb          5.9kb
green  open   .kibana_task_manager           VXfWHMtISMeKM8nSyZgyKA   1   1          2            0     25.2kb         12.6kb
green  open   elastalert_status_silence      Oj21DSvQRUqq2X2ACJ4bGg   5   1          6            0     45.6kb         22.8kb
```

Step 3:- Configure The global Elastalert configuration

- Rename the config.yaml.example file to config.yaml
- Provide the instance informations like host,port,ssl_required,username,password,etc..

Step 4:- Dump some data to the Elasticsearch

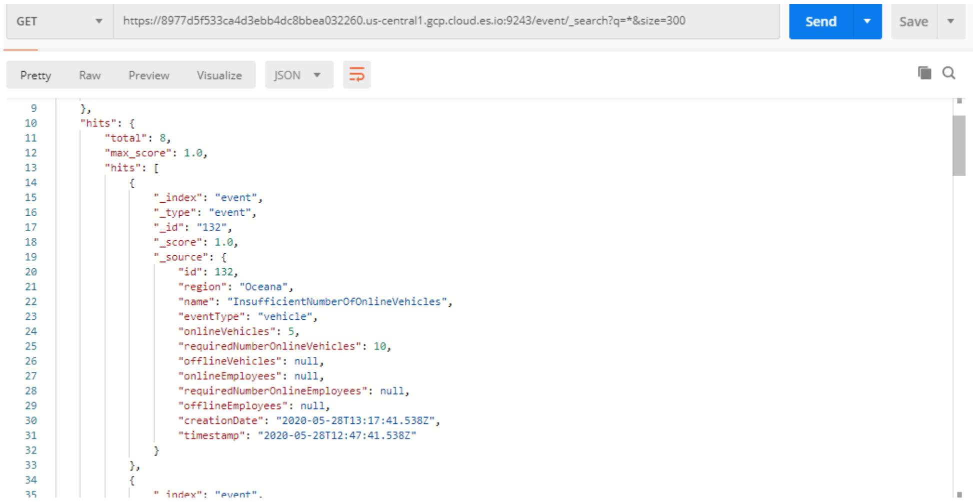
- Create a spring boot application from initializer
- You can connect with Elasticsearch either by using Spring data jest/Spring data Elasticsearch
    - Using Spring Data Jest use (Deprecated)
        - Spring boot version---> 2.1.11.RELEASE
        - Spring data Jest version ---> 3.2.2.RELEASE
        - Required properties:-
          spring.data.elasticsearch.properties.path.logs=target/elasticsearch/log
          spring.data.elasticsearch.properties.path.data=target/elasticsearch/data
          spring.data.elasticsearch.cluster-name=# clustername
          spring.data.elasticsearch.cluster-nodes=#host:port
          spring.data.jest.password=#password
          spring.data.jest.username=#username
          spring.data.jest.readTimeout=100000000
          spring.data.jest.uri=#https://host:port
    - Using Spring Data Elasticsearch
        - You can use the latest upgrade 4.0 with spring boot 2.3.0
        - Required properties:-
          spring.data.elasticsearch.cluster-name=elasticsearch # Elasticsearch cluster name.
          spring.data.elasticsearch.cluster-nodes= # Comma-separated list of cluster node addresses.
          spring.data.elasticsearch.properties.*= # Additional properties used to configure the client.
          spring.data.elasticsearch.repositories.enabled=true # Whether to enable Elasticsearch repositories

Important Note :- The data we dump into an index need a date field with **ISO8601** or **Unix timestamped** format
In spring boot we can acheive this by adding :-
@Field(type = FieldType.Date, store = true, format = DateFormat.custom, pattern = "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'")
@JsonFormat (shape = JsonFormat.Shape.STRING, pattern ="yyyy-MM-dd'T'HH:mm:ss.SSS'Z'")
private Date timestamp;

Timestamp mapping should look like this in the index:-

```
 "timestamp": {
                    "type": "date",
                    "store": true,
                    "format": "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
                }
```

Example, Create an index with name Event and dump some data and read using https://host:port/event/_search?q=*



https://8977d5f533ca4d3ebb4dc8bbea032260.us-central1.gcp.cloud.es.io:9243/event/_search?q=*&size=300

```
    },
    "hits": {
        "total": 8,
        "max_score": 1.0,
        "hits": [
            {
                "_index": "event",
                "_type": "event",
                "_id": "132",
                "_score": 1.0,
                "_source": {
                    "id": 132,
                    "region": "Oceana",
                    "name": "InsufficientNumberOfOnlineVehicles",
                    "eventType": "vehicle",
                    "onlineVehicles": 5,
                    "requiredNumberOnlineVehicles": 10,
                    "offlineVehicles": null,
                    "onlineEmployees": null,
                    "requiredNumberOnlineEmployees": null,
                    "offlineEmployees": null,
                    "creationDate": "2020-05-28T13:17:41.538Z",
                    "timestamp": "2020-05-28T12:47:41.538Z"
                }
            },
            {
                "_index": "event",
```

Step 5:- Create Alert Rules and configure Alert Types

- Create rules

There are different types of rules you can create like any,new_term, blacklist, whitelist, frequency etc..,
Refer this for creating a rule --> https://elastalert.readthedocs.io/en/latest/running_elastalert.html#creating-a-rule
Refer this for rules configurations and types --> https://elastalert.readthedocs.io/en/latest/ruletypes.html
Refer this to know available alerts that you can configure with the alert ---> https://elastalert.readthedocs.io/en/latest/ruletypes.html#alerts

- Timestamp configuration

If the timestamp is not a @timestamp you need to explicitly say which field to use as timestamp, provide the below configurations to the rule YAML file:-
timestamp_field: timestamp
timestamp_type: custom
timestamp_format: '%Y-%m-%dT%H:%M:%S.%fZ'
timestamp_format_expr: 'ts[:23] + ts[26:]

- Email SMTP Configuration for rules

alert:
- "email"
email:
- "xxx.s@xxx.com" (to list)
email_from_field: "alerts"
email_add_domain: "@ucc.com"
smtp_host: "smtp.gmail.com"
smtp_port: 465
smtp_ssl: true
smtp_auth_file: 'email_auth.yaml' (Authentication credentials file)

The email_auth.yaml should contain username and password in YAML format for example:-

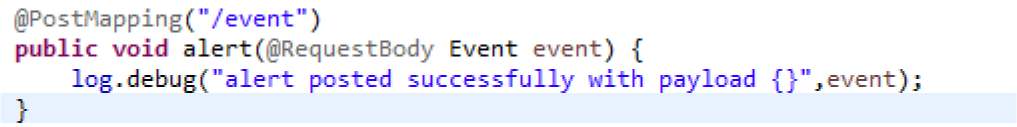
user: xxxx@gmail.com
password: xxxxx

- HTTP POST Alert Type

This will post the resultant alert as JSON, the required configurations are

alert: post
http_post_url: "http://localhost:8080/api/event"

In our server, we need to have a rest endpoint which accepts the JSON body



```
@PostMapping("/event")
public void alert(@RequestBody Event event) {
    log.debug("alert posted successfully with payload {}",event);
}
```

Example Rules

blacklist.yaml	email_auth.yaml

events_term.yaml

change_rule_http_post.yaml

frequency_rule.yaml

flatline.yaml

any_rule.yaml

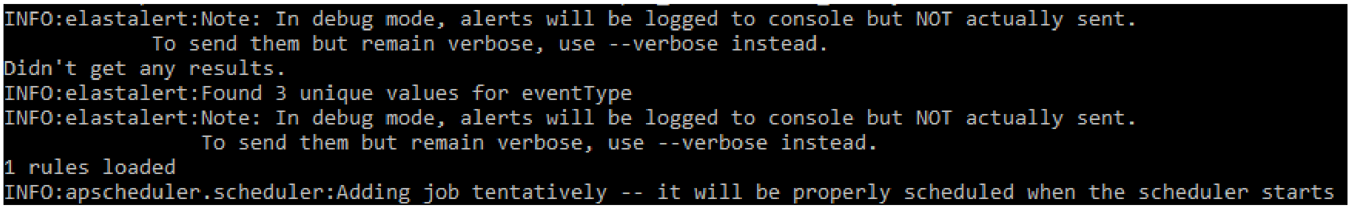
whitelist.yaml

Step 7:- Testing the alert rule

Use this command :- **elastalert-test-rule** your_rules/your_rule.yaml
This is only for test purpose this will log the details to the console, this will not send actual alerts if we want to send the real alerts we can specify the **--alert** flag, below are output logs of some rule types

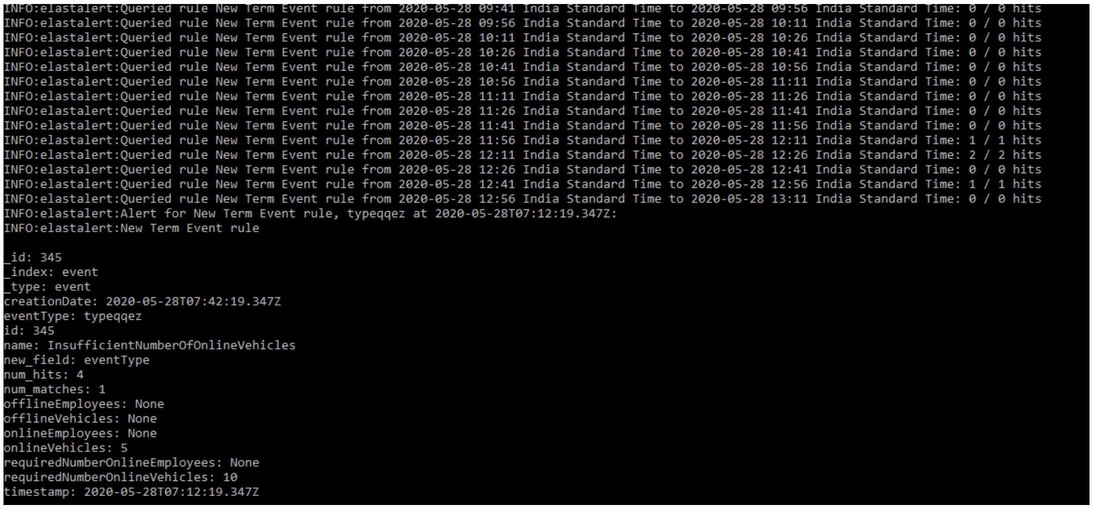
New Term (Email alert)

- This rule matches when a new value appears in a field that has never been seen before. When ElastAlert starts, it will use an aggregation query to gather all known terms for a list of fields/query_key



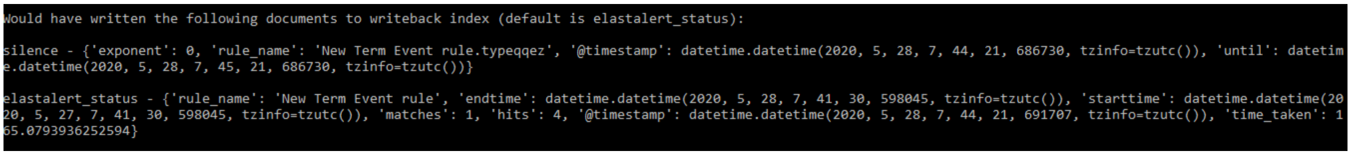
```
INFO:elastalert:Note: In debug mode, alerts will be logged to console but NOT actually sent.
                To send them but remain verbose, use --verbose instead.
Didn't get any results.
INFO:elastalert:Found 3 unique values for eventType
INFO:elastalert:Note: In debug mode, alerts will be logged to console but NOT actually sent.
                     To send them but remain verbose, use --verbose instead.
1 rules loaded
INFO:apscheduler.scheduler:Adding job tentatively -- it will be properly scheduled when the scheduler starts
```

- Elastalert will poll the Elasticsearch according to the configured buffer_time and print the matched document to the console



```
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 09:41 India Standard Time to 2020-05-28 09:56 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 09:56 India Standard Time to 2020-05-28 10:11 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 10:11 India Standard Time to 2020-05-28 10:26 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 10:26 India Standard Time to 2020-05-28 10:41 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 10:41 India Standard Time to 2020-05-28 10:56 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 10:56 India Standard Time to 2020-05-28 11:11 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 11:11 India Standard Time to 2020-05-28 11:26 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 11:26 India Standard Time to 2020-05-28 11:41 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 11:41 India Standard Time to 2020-05-28 11:56 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 11:56 India Standard Time to 2020-05-28 12:11 India Standard Time: 1 / 1 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 12:11 India Standard Time to 2020-05-28 12:26 India Standard Time: 2 / 2 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 12:26 India Standard Time to 2020-05-28 12:41 India Standard Time: 0 / 0 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 12:41 India Standard Time to 2020-05-28 12:56 India Standard Time: 1 / 1 hits
INFO:elastalert:Queried rule New Term Event rule from 2020-05-28 12:56 India Standard Time to 2020-05-28 13:11 India Standard Time: 0 / 0 hits
INFO:elastalert:Alert for New Term Event rule, typeqqez at 2020-05-28T07:12:19.347Z:
INFO:elastalert:New Term Event rule

_id: 345
_index: event
_type: event
creationDate: 2020-05-28T07:42:19.347Z
eventType: typeqqez
id: 345
name: InsufficientNumberOfOnlineVehicles
new_field: eventType
num_hits: 4
num_matches: 1
offlineEmployees: None
offlineVehicles: None
onlineEmployees: None
onlineVehicles: 5
requiredNumberOnlineEmployees: None
requiredNumberOnlineVehicles: 10
timestamp: 2020-05-28T07:12:19.347Z
```

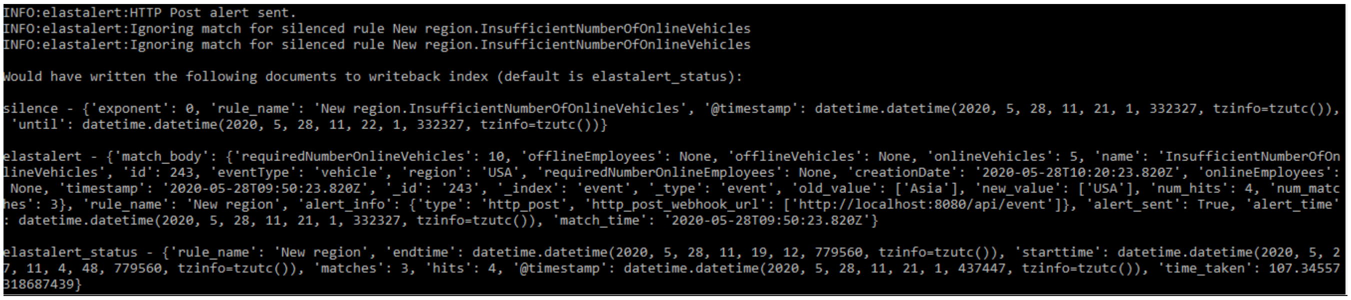
- At last, the server will write the results to the index for audit purpose



```
Would have written the following documents to writeback index (default is elastalert_status):

silence - {'exponent': 0, 'rule_name': 'New Term Event rule.typeqqez', '@timestamp': datetime.datetime(2020, 5, 28, 7, 44, 21, 686730, tzinfo=tzutc()), 'until': datetim
e.datetime(2020, 5, 28, 7, 45, 21, 686730, tzinfo=tzutc())}

elastalert_status - {'rule_name': 'New Term Event rule', 'endtime': datetime.datetime(2020, 5, 28, 7, 41, 30, 598045, tzinfo=tzutc()), 'starttime': datetime.datetime(20
20, 5, 27, 7, 41, 30, 598045, tzinfo=tzutc()), 'matches': 1, 'hits': 4, '@timestamp': datetime.datetime(2020, 5, 28, 7, 44, 21, 691707, tzinfo=tzutc()), 'time_taken': 1
65.0793936252594}
```

Change (HTTP POST Alert)

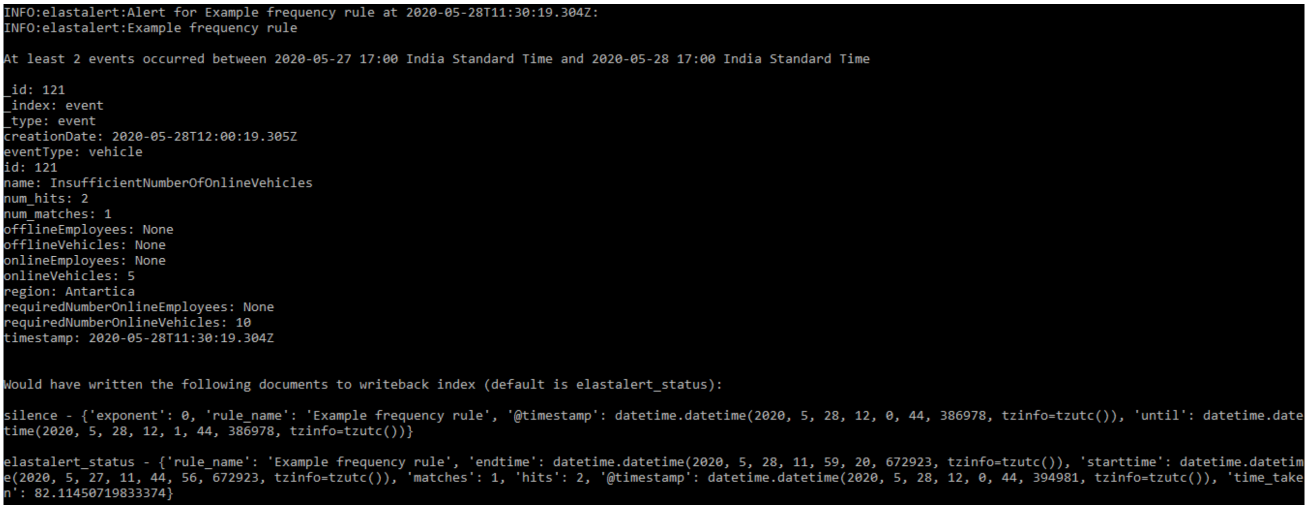
- This rule will monitor a certain field and match if that field changes
- Resultant log at the end (real alerts are sending used --alert flag)



```
INFO:elastalert:HTTP Post alert sent.
INFO:elastalert:Ignoring match for silenced rule New region.InsufficientNumberOfOnlineVehicles
INFO:elastalert:Ignoring match for silenced rule New region.InsufficientNumberOfOnlineVehicles

Would have written the following documents to writeback index (default is elastalert_status):

silence - {'exponent': 0, 'rule_name': 'New region.InsufficientNumberOfOnlineVehicles', '@timestamp': datetime.datetime(2020, 5, 28, 11, 21, 1, 332327, tzinfo=tzutc()),
 'until': datetime.datetime(2020, 5, 28, 11, 22, 1, 332327, tzinfo=tzutc())}

elastalert - {'match_body': {'requiredNumberOnlineVehicles': 10, 'offlineEmployees': None, 'offlineVehicles': None, 'onlineVehicles': 5, 'name': 'InsufficientNumberOfOn
lineVehicles', 'id': 243, 'eventType': 'vehicle', 'region': 'USA', 'requiredNumberOnlineEmployees': None, 'creationDate': '2020-05-28T10:20:23.820Z', 'onlineEmployees':
 None, 'timestamp': '2020-05-28T09:50:23.820Z', '_id': '243', '_index': 'event', '_type': 'event', 'old_value': ['Asia'], 'new_value': ['USA'], 'num_hits': 4, 'num_matc
hes': 3}, 'rule_name': 'New region', 'alert_info': {'type': 'http_post', 'http_post_webhook_url': ['http://localhost:8080/api/event']}, 'alert_sent': True, 'alert_time'
: datetime.datetime(2020, 5, 28, 11, 21, 1, 332327, tzinfo=tzutc()), 'match_time': '2020-05-28T09:50:23.820Z'}

elastalert_status - {'rule_name': 'New region', 'endtime': datetime.datetime(2020, 5, 28, 11, 19, 12, 779560, tzinfo=tzutc()), 'starttime': datetime.datetime(2020, 5, 2
7, 11, 4, 48, 779560, tzinfo=tzutc()), 'matches': 3, 'hits': 4, '@timestamp': datetime.datetime(2020, 5, 28, 11, 21, 1, 437447, tzinfo=tzutc()), 'time_taken': 107.34557
318687439}
```

Frequency

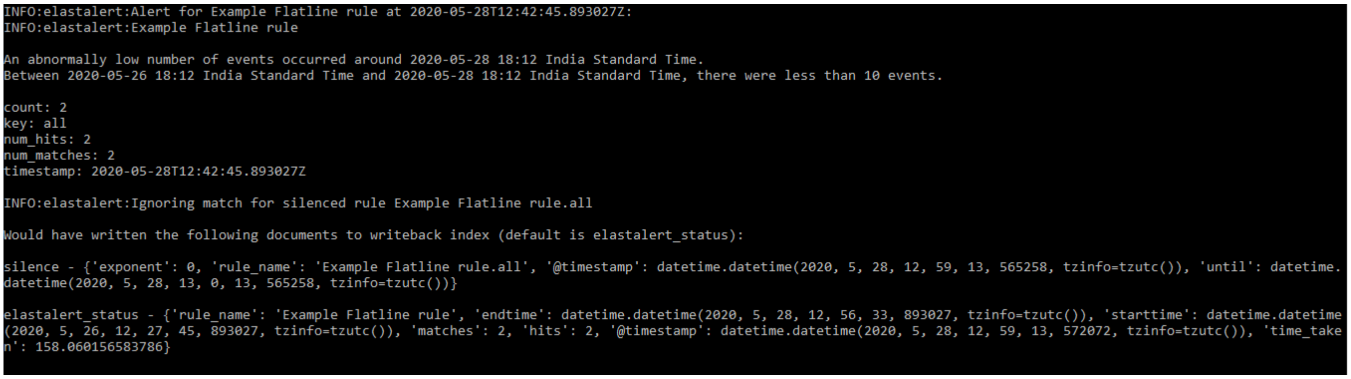
- This rule matches when there are at least a certain number of events in a given time frame.
- Resultant log at the end



```
INFO:elastalert:Alert for Example frequency rule at 2020-05-28T11:30:19.304Z:
INFO:elastalert:Example frequency rule

At least 2 events occurred between 2020-05-27 17:00 India Standard Time and 2020-05-28 17:00 India Standard Time

_id: 121
_index: event
_type: event
creationDate: 2020-05-28T12:00:19.305Z
eventType: vehicle
id: 121
name: InsufficientNumberOfOnlineVehicles
num_hits: 2
num_matches: 1
offlineEmployees: None
offlineVehicles: None
onlineEmployees: None
onlineVehicles: 5
region: Antartica
requiredNumberOnlineEmployees: None
requiredNumberOnlineVehicles: 10
timestamp: 2020-05-28T11:30:19.304Z


Would have written the following documents to writeback index (default is elastalert_status):

silence - {'exponent': 0, 'rule_name': 'Example frequency rule', '@timestamp': datetime.datetime(2020, 5, 28, 12, 0, 44, 386978, tzinfo=tzutc()), 'until': datetime.date
time(2020, 5, 28, 12, 1, 44, 386978, tzinfo=tzutc())}

elastalert_status - {'rule_name': 'Example frequency rule', 'endtime': datetime.datetime(2020, 5, 28, 11, 59, 20, 672923, tzinfo=tzutc()), 'starttime': datetime.datetim
e(2020, 5, 27, 11, 44, 56, 672923, tzinfo=tzutc()), 'matches': 1, 'hits': 2, '@timestamp': datetime.datetime(2020, 5, 28, 12, 0, 44, 394981, tzinfo=tzutc()), 'time_take
n': 82.11450719833374}
```

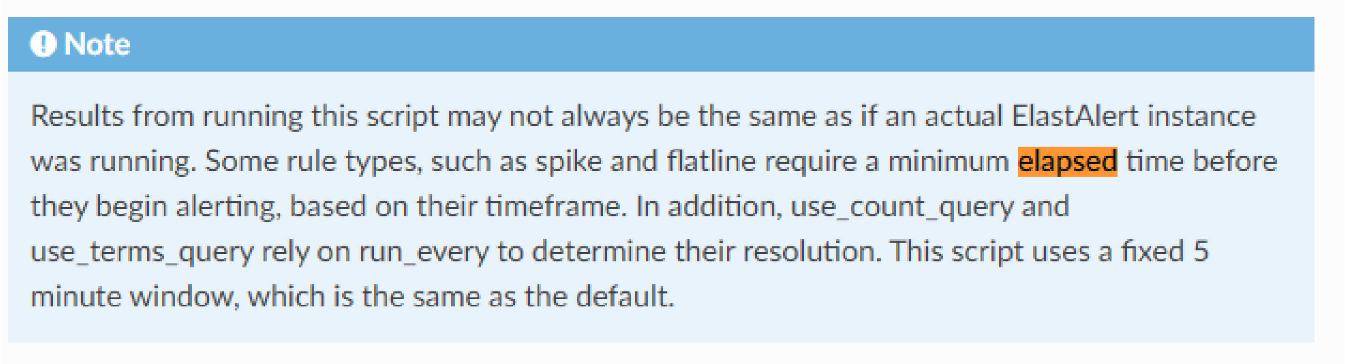
Flatline

- This rule matches when the total number of events is under a given `threshold` for a time period.
- Resultant log at the end



```
INFO:elastalert:Alert for Example Flatline rule at 2020-05-28T12:42:45.893027Z:
INFO:elastalert:Example Flatline rule

An abnormally low number of events occurred around 2020-05-28 18:12 India Standard Time.
Between 2020-05-26 18:12 India Standard Time and 2020-05-28 18:12 India Standard Time, there were less than 10 events.

count: 2
key: all
num_hits: 2
num_matches: 2
timestamp: 2020-05-28T12:42:45.893027Z

INFO:elastalert:Ignoring match for silenced rule Example Flatline rule.all

Would have written the following documents to writeback index (default is elastalert_status):

silence - {'exponent': 0, 'rule_name': 'Example Flatline rule.all', '@timestamp': datetime.datetime(2020, 5, 28, 12, 59, 13, 565258, tzinfo=tzutc()), 'until': datetime.
datetime(2020, 5, 28, 13, 0, 13, 565258, tzinfo=tzutc())}

elastalert_status - {'rule_name': 'Example Flatline rule', 'endtime': datetime.datetime(2020, 5, 28, 12, 56, 33, 893027, tzinfo=tzutc()), 'starttime': datetime.datetime
(2020, 5, 26, 12, 27, 45, 893027, tzinfo=tzutc()), 'matches': 2, 'hits': 2, '@timestamp': datetime.datetime(2020, 5, 28, 12, 59, 13, 572072, tzinfo=tzutc()), 'time_take
n': 158.060156583786}
```

When using flatline rule or spike this won't work as we expected in a test environment, this needs a minimum_elapsed_timeframe to begin the alerts



Note

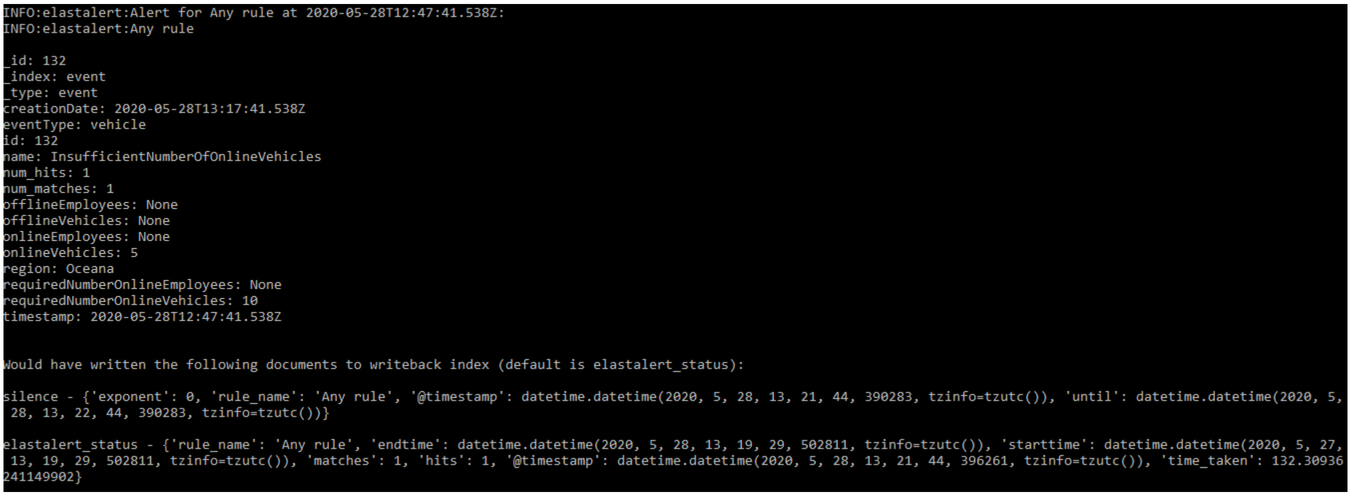
Results from running this script may not always be the same as if an actual ElastAlert instance was running. Some rule types, such as spike and flatline require a minimum elapsed time before they begin alerting, based on their timeframe. In addition, use_count_query and use_terms_query rely on run_every to determine their resolution. This script uses a fixed 5 minute window, which is the same as the default.

In order to trigger the alerts, we can increase the timeframe with the configuration

timeframe:
days: 2

Any

- Any rule will match everything. Every hit that the query returns will generate an alert.
- Resultant log at the end



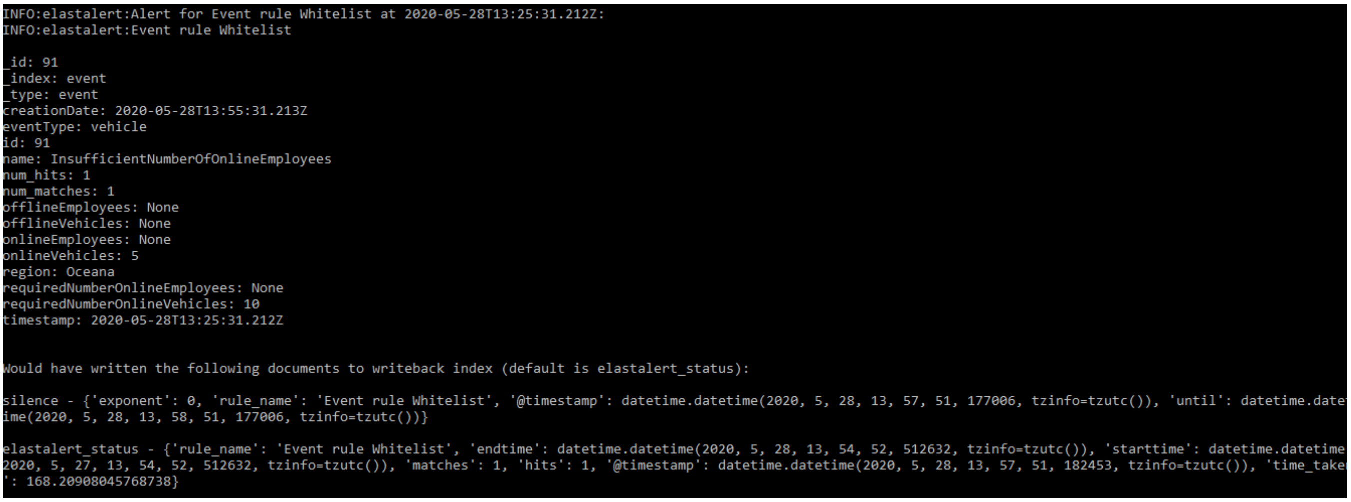
```
INFO:elastalert:Alert for Any rule at 2020-05-28T12:47:41.538Z:
INFO:elastalert:Any rule

_id: 132
_index: event
_type: event
creationDate: 2020-05-28T13:17:41.538Z
eventType: vehicle
id: 132
name: InsufficientNumberOfOnlineVehicles
num_hits: 1
num_matches: 1
offlineEmployees: None
offlineVehicles: None
onlineEmployees: None
onlineVehicles: 5
region: Oceana
requiredNumberOnlineEmployees: None
requiredNumberOnlineVehicles: 10
timestamp: 2020-05-28T12:47:41.538Z


Would have written the following documents to writeback index (default is elastalert_status):

silence - {'exponent': 0, 'rule_name': 'Any rule', '@timestamp': datetime.datetime(2020, 5, 28, 13, 21, 44, 390283, tzinfo=tzutc()), 'until': datetime.datetime(2020, 5,
 28, 13, 22, 44, 390283, tzinfo=tzutc())}

elastalert_status - {'rule_name': 'Any rule', 'endtime': datetime.datetime(2020, 5, 28, 13, 19, 29, 502811, tzinfo=tzutc()), 'starttime': datetime.datetime(2020, 5, 27,
 13, 19, 29, 502811, tzinfo=tzutc()), 'matches': 1, 'hits': 1, '@timestamp': datetime.datetime(2020, 5, 28, 13, 21, 44, 396261, tzinfo=tzutc()), 'time_taken': 132.30936
241149902}
```

Blacklist

- The blacklisting rule will check a certain field against a blacklist, and match if it is in the blacklist.

Whitelist

- Similar to `blacklist`, this rule will compare a certain field to a whitelist, and match if the list does not contain the term.
- Apart from the blacklist, the whitelist had a required field `ignore_null`: If true, events without a `compare_key` field will not match. If not provided a validation error will throw
- Resultant log at the end



```
INFO:elastalert:Alert for Event rule Whitelist at 2020-05-28T13:25:31.212Z:
INFO:elastalert:Event rule Whitelist

_id: 91
_index: event
_type: event
creationDate: 2020-05-28T13:55:31.213Z
eventType: vehicle
id: 91
name: InsufficientNumberOfOnlineEmployees
num_hits: 1
num_matches: 1
offlineEmployees: None
offlineVehicles: None
onlineEmployees: None
onlineVehicles: 5
region: Oceana
requiredNumberOnlineEmployees: None
requiredNumberOnlineVehicles: 10
timestamp: 2020-05-28T13:25:31.212Z


Would have written the following documents to writeback index (default is elastalert_status):

silence - {'exponent': 0, 'rule_name': 'Event rule Whitelist', '@timestamp': datetime.datetime(2020, 5, 28, 13, 57, 51, 177006, tzinfo=tzutc()), 'until': datetime.date
ime(2020, 5, 28, 13, 58, 51, 177006, tzinfo=tzutc())}

elastalert_status - {'rule_name': 'Event rule Whitelist', 'endtime': datetime.datetime(2020, 5, 28, 13, 54, 52, 512632, tzinfo=tzutc()), 'starttime': datetime.datetime
2020, 5, 27, 13, 54, 52, 512632, tzinfo=tzutc()), 'matches': 1, 'hits': 1, '@timestamp': datetime.datetime(2020, 5, 28, 13, 57, 51, 182453, tzinfo=tzutc()), 'time_take
': 168.20908045768738}
```

<u>Common Issues/concerns might face during the test</u>

- Configure Buffer Time

Elastalert checks the data in a range of 15 minutes(comparing with the timestamp we are provided to the document) we can customize this by passing --start and --end parameters

We can customize the polling time using "**buffer_time**" By default, ElastAlert will query large overlapping windows to ensure that it does not miss any events, even if they are indexed in real-time. In config.YAML, you can adjust "buffer_time to a smaller number to only query the most recent few minutes.

buffer_time:
    minutes: 5

- Why did I only get one alert when I expected to get several?

Log:- "INFO:elastalert: Ignoring match for silenced rule Event rule"
There is a setting called **realert** which is the minimum time between two alerts for the same rule. Any alert that occurs within this time will simply be dropped. The default value for this is one minute. If you want to receive an alert for every single match, even if they occur right after each other, use, by default this property is 1

```
realert:
  minutes: 0
```

- Issue With keyword type fields

Log: INFO:elastalert:Found no values for your_field (When pulling the data for the first time)

When you're working with term queries, for example writing a **new_term** rule, by default the string fields mapping type will be the **keyword**, So when you specify a field for filter/querying you need to postfix **.keyword** to the fields. Elastalert provides **use_keyword_postfix** boolean configuration to solve this issue, by default this property is set to **true**. another solution here is to change the field type itself in Elasticsearch you can do this by annotating the field **@ Field(type = FieldType.Text, fielddata = true)**

To see the mapping for the index:- https://host:port/index/_mapping/

For a String field by default, the mapping will be

```
 "name": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
```

A Very helpful **resource**:- https://awesomeopensource.com/project/Yelp/elastalert

- Issue when testing the HTTP Post Alert type

Issue:- AttributeError: '_thread._local' object has no attribute 'alerts_sent'

Fix:- Open elatalert.py on the local AppData, make the below change

```
self.replace_dots_in_field_names = self.conf.get('replace_dots_in_field_names', False)
self.thread_data.num_hits = 0
self.thread_data.num_dupes = 0
self.thread_data.alerts_sent = 0 #add
self.scheduler = BackgroundScheduler()
```

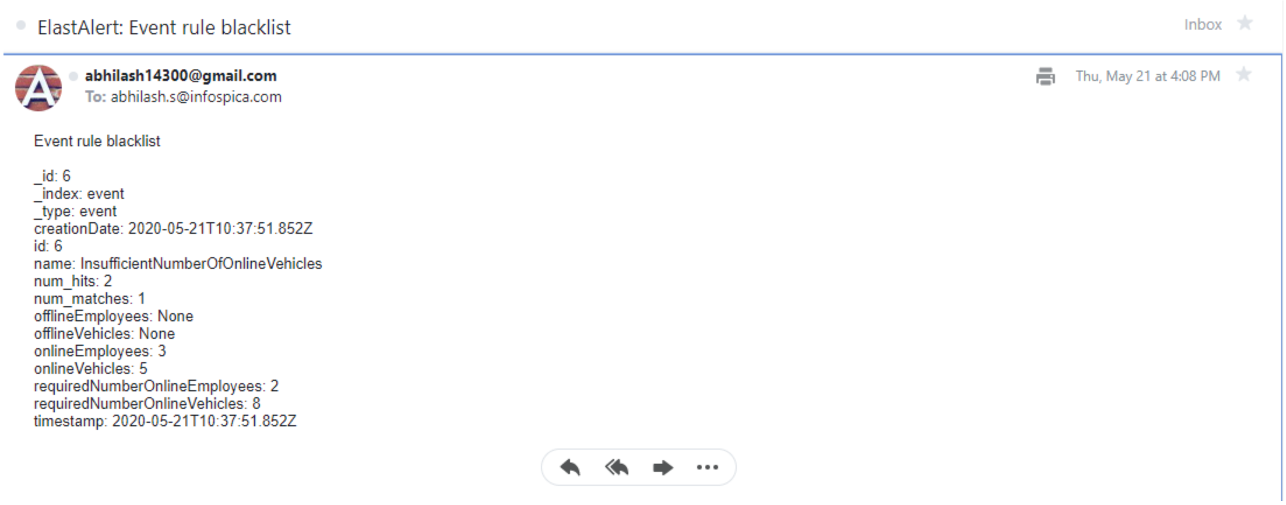
Step 8:- Run the Elastalert server

- Use this command:- **python -m elastalert.elastalert --verbose --rule your_rules_dir/your_rule.yaml**
- Output Log

```
INFO:elastalert:Queried rule Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:05 India Standard Time: 0 / 0 hits
INFO:elastalert:Ran Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:05 India Standard Time: 0 query hits (0 already seen), 0 matches, 0 alerts sent
INFO:elastalert:Disabled rules are: []
INFO:elastalert:Sleeping for 59.9995 seconds
INFO:elastalert:Background configuration change check run at 2020-05-21 16:06 India Standard Time
INFO:elastalert:Sent email to ['abhilash.s@infospica.com']
INFO:elastalert:Sent email to ['abhilash.s@infospica.com']
INFO:elastalert:Queried rule Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:06 India Standard Time: 1 / 1 hits
INFO:elastalert:Background alerts thread 2 pending alerts sent at 2020-05-21 16:06 India Standard Time
INFO:elastalert:Sent email to ['abhilash.s@infospica.com']
INFO:elastalert:Ran Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:06 India Standard Time: 1 query hits (0 already seen), 1 matches, 1 alerts sent
INFO:elastalert:Disabled rules are: []
INFO:elastalert:Sleeping for 60.0 seconds
INFO:elastalert:Background configuration change check run at 2020-05-21 16:07 India Standard Time
INFO:elastalert:Background alerts thread 0 pending alerts sent at 2020-05-21 16:07 India Standard Time
INFO:elastalert:Queried rule Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:07 India Standard Time: 1 / 1 hits
INFO:elastalert:Ran Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:07 India Standard Time: 1 query hits (1 already seen), 0 matches, 0 alerts sent
INFO:elastalert:Disabled rules are: []
INFO:elastalert:Sleeping for 60.0 seconds
INFO:elastalert:Background configuration change check run at 2020-05-21 16:08 India Standard Time
INFO:elastalert:Background alerts thread 0 pending alerts sent at 2020-05-21 16:08 India Standard Time
INFO:elastalert:Queried rule Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:08 India Standard Time: 2 / 2 hits
INFO:elastalert:Sent email to ['abhilash.s@infospica.com']
INFO:elastalert:Ran Event rule blacklist from 2020-05-21 16:02 India Standard Time to 2020-05-21 16:08 India Standard Time: 2 query hits (1 already seen), 1 matches, 1 alerts sent
INFO:elastalert:SIGINT received, stopping ElastAlert...
```

- Email Message:-

The email alert looks like this (We can customize the subject and content)



ElastAlert: Event rule blacklist

abhilash14300@gmail.com
To: abhilash.s@infospica.com

Event rule blacklist

```
_id: 6
_index: event
_type: event
creationDate: 2020-05-21T10:37:51.852Z
id: 6
name: InsufficientNumberOfOnlineVehicles
num_hits: 2
num_matches: 1
offlineEmployees: None
offlineVehicles: None
onlineEmployees: 3
onlineVehicles: 5
requiredNumberOnlineEmployees: 2
requiredNumberOnlineVehicles: 8
timestamp: 2020-05-21T10:37:51.852Z
```

ⓘ

Related articles

Content by label

There is no content with the specified labels