Watcher

เป็น plugin ใน elasticsearch ไว้ดูAction ต่างๆ แล้วสามารถแจ้งเตือนได้
เข้าไปที่เครื่อง elasticsearch 1
ip: 52.74.104.59
Key: TMN-Log-Monitoring-key-pair

ใช้ คำสั่ง sudo su เพื่อเข้าสิทธิ root

Install Watcher

1. cd ไปที่ usr/share/elasticsearch/ เพื่อติดตั้งLicense plugin โดยใช้คำสั่ง
   bin/plugin -i elasticsearch/license/latest
2. ติดตั้ง Watcher plugin โดยใช้คำสั่ง
   bin/plugin -i elasticsearch/watcher/latest

ดูเพิ่มเติมที่
https://www.elastic.co/guide/en/watcher/current/getting-started.html

---

Configuring Watcher to Send Email

cd ไปที่ etc/elasticsearch/ ใช้คำสั่ง vi elasticsearch.yml
เพื่อเพิ่มคำสั่งลงในไฟล์ elasticsearch.yml

1. ถ้าใช้ Gmail (Sending Email from Gmail)

```
watcher.actions.email.service.account:
    gmail_account:
        profile: gmail
        smtp:
            auth: true
            starttls.enable: true
            host: smtp.gmail.com
            port: 587
            user: <username>
            password: <password>
```

2. ถ้าใช้ Outlook (Sending Email from Outlook)

```
watcher.actions.email.service.account:
    outlook_account:
        profile: outlook
        smtp:
            auth: true
            starttls.enable: true
            host: smtp-mail.outlook.com
            port: 587
            user: <username>
            password: <password>
```

3. ถ้าใช้ Exchange (Sending Email from Exchange)

```
watcher.actions.email.service.account:
  exchange_account:
    profile: outlook
    email_defaults:
      from: <email address of service account> (1)
    smtp:
      auth: true
      starttls.enable: true
      host: <your exchange server>
      port: 587
      user: <email address of service account> (2)
      password: <password>
```

(1) Some organizations configure Exchange to validate that the `from` field is a valid local email account.

(2) Many organizations support use of your email address as your username, though it is a good idea to check with your system administrator if you receive authentication-related failures.

4. ถ้าใช้ Amazon SES (Sending Email from Amazon SES)

```
watcher.actions.email.service.account:
   ses_account:
      smtp:
         auth: true
         starttls.enable: true
         starttls.required: true
         host: email-smtp.us-east-1.amazonaws.com (1)
         port: 587
         user: <username>
         password: <password>
```

(1) smtp.host varies depending on the region

Table 1. Email Account Attributes

Name	Required	Default	Description
profile	no	standard	The profile to use to build the MIME messages that are sent from the account. Valid values: standard (default),gmail and outlook.
email_defaults.*	no	-	An optional set of email attributes to use as defaults for the emails sent from the account. See Email Action Attributes for the supported attributes. for the possible email attributes)
smtp.auth	no	false	When true, attempt to authenticate the user using the AUTH command.
smtp.host	yes	-	The SMTP server to connect to.

smtp.port	no	25	The SMTP server port to connect to.
smtp.user	yes	-	The user name for SMTP.
smtp.password	no	-	The password for the specified SMTP user.
smtp.starttls.enable	no	false	When true, enables the use of the STARTTLScommand (if supported by the server) to switch the connection to a TLS-protected connection before issuing any login commands. Note that an appropriate trust store must configured so that the client will trust the server's certificate. Defaults to false.
smtp.*	no	-	SMTP attributes that enable fine control over the SMTP protocol when sending messages. Seecom.sun.mail.smtp for the full list of SMTP properties you can set.

ตัวอย่างการใช้งาน : ใช้ Gmail ในการส่ง

```
watcher.actions.email.service.account:
   gmail_account:
      profile: gmail
      smtp:
         auth: true
         starttls.enable: true
         host: smtp.gmail.com
         port: 587
         user: scouter.ascend@gmail.com
         password: sprintfail
```

ดูเพิ่มเติมที่
https://www.elastic.co/guide/en/watcher/current/email-services.html

---

คำสั่งในการค้นหากำหนด Action และส่งEmailให้Email Address ที่ต้องการส่ง

ใน '.....' เป็นการเขียนโดยใช้JSON

ตัวอย่าง

```
curl -u watcher:watcher -XPUT
'http://localhost:9200/_watcher/watch/kiosk_timeout' -d '{---0
 "trigger" :{ "schedule" : {"hourly" : { "minute" : 0 } }},   ---1
 "input" : {
  "search" : {
   "request" : {
    "indices" : [ "kiosk-*" ], ---2
    "body" : {
     "query" : {
      "match" : { "message": "java.net.SocketTimeoutException" } ---3
     },
     "filter" : {
        "range" : {"@timestamp" : {"gte" : "now-1h"}} ---4
     }
    }
   }
  }
 },
 "condition" : {
  "compare" : {
   "ctx.payload.hits.total" : { "gte" : 10 } ---5
   }
 },
 "actions" : {
  "email_timeout_kiosk" : { ---6
   "email" : {
    "to" : "scouter.ascend@gmail.com", ---7
    "subject" : "you have {{ctx.payload.hits.total}}
java.net.SocketTimeoutException ", ---8
     "body" : "pleace, check your log", ---9
      "attach_data" : true, ---10
      "priority" : "high" ---11
   }
  }
 }
}'
```

0. -u watcher:watcher -XPUT 'http://localhost:9200/_watcher/watch/kiosk_timeout' -d '{.....}'
-u watcher:watcher เป็นการใส่ username:password
-XPUT คือคำสั่งInput
'http://localhost:9200/_watcher/watch/kiosk_timeout' ไปสั่งที่localhost โดยใช้ port:9200
โดย kiosk_timeout คือชื่อของ watcher ที่ดูAction นี้
-d'{.....}' ข้างใน'{}' คือการกระทำที่กำหนด

1. "trigger" :{ "schedule" : {"hourly" : { "minute" : 0 } }},
#เป็นการให้ watcher ไปดูตามเวลาที่กำหนด
ตัวอย่างเป็นการดูทุกชั่วโมง ในเวลา 0 นาที เช่น 9.00 , 12.00, 15.00
เราสามารถเพิ่มให้ดูทุก15 นาที เช่น 9.15, 9.30 , 9.45
เราสามารถตั้งเวลาได้หลายแบบ ดูเพิ่มเติมที่
https://www.elastic.co/guide/en/watcher/current/trigger.html

2. "indices" : ["kiosk-*"],
#เป็นการให้ไปดูที่ indices ชื่อ "kiosk-" แล้วต่อด้วยอะไรก็ได้

3. "query" : {
"match" : { "message": "java.net.SocketTimeoutException" } },
#เป็นการไป query tag ชื่อ "message"
ค้นคำว่า "java.net.SocketTimeoutException" #เนื่องความสามรถของ Elasticsearch คือ full text search ทำให้ค้นหาคำนี้ได้เลย
ดูเพิ่มเติมที่
https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html

4. "filter" : { "range" : {"@timestamp" : {"gte" : "now-1h"}}}
#เป็นการกำหนดช่วงเวลาที่ให้ watcher ย้อนกลับไปดู โดยใช้
tag "@timestamp" เป็นตัวเปรียบเทียบเวลา
"gte" คือบอกว่า มากกว่าเท่ากับ
"now-1h" คือเวลาปัจจุบัน ย้อนกลับไป 1 ชั่วโมง
ดูเพิ่มเติมที่
https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-range-filter.html
และ
https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-filtered-query.html

5. "condition" : {"compare" : { "ctx.payload.hits.total" : { "gte" : 10 } } },
#เป็นการกำหนดเงื่อนไข จากตัวอย่าง จะใช้
"compare" เปรียบเทียบ
"ctx.payload.hits.total" คือ จำนวนpayload ทั้งหมด
"gte" คือบอกว่า มากกว่าเท่ากับ ในตัวอย่างคือ ถ้ามากกว่าเท่ากับ 10
ดูเพิ่มเติมที่
https://www.elastic.co/guide/en/watcher/current/condition.html

6. "email_timeout_kiosk" #คือ ID ของ Action สามารถเปลี่ยนชื่อได้

7. "to" : "scouter.ascend@gmail.com", #Email ที่ต้องการจะส่ง

8. "subject" : "you have {{ctx.payload.hits.total}}
java.net.SocketTimeoutException ",
#เป็นการเขียน subject ของ Email
{{ctx.payload.hits.total}} จะบอกจำนวนpayload ทั้งหมดในเวลาที่กำหนด

9. "body" : "pleace, check your log",
#เป็นส่วนเนื้อหาที่เราต้องการจะแจ้ง

10. "attach_data" : true,
#เป็นการให้แนบไฟล์log ที่เราต้องการให้แจ้ง

11. "priority" : "high"
#เป็นการกำหนด priority ให้กับ action นี้
**Action ทั้งหมดจะถูกเก็บไว้ที่ indices .watch_history-"ปี/เดือน/วัน"
เช่น **.watch_history-2015.07.16**
ดูเพิ่มเติมที่
https://www.elastic.co/guide/en/watcher/current/actions.html

การยกเลิกคำสั่ง Action

curl -u watcher:watcher -XDELETE
'http://localhost:9200/_watcher/watch/kiosk_timeout'
-u watcher:watcher เป็นการใส่ username:password
-XDELETE คือคำสั่งในการลบAction นี้
'http://localhost:9200/_watcher/watch/kiosk_timeout' ไปสั่งที่localhost
โดยใช้ port:9200
โดย kiosk_timeout คือชื่อของ watcher ที่ดูAction นี้

Reference:
https://www.elastic.co/guide/en/watcher/current/index.html