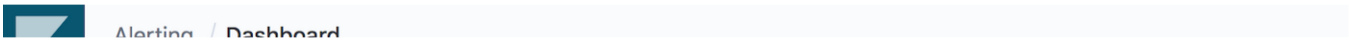
Monitor and Alert from Elasticsearch and Kibana, in AWS Managed Elasticsearch

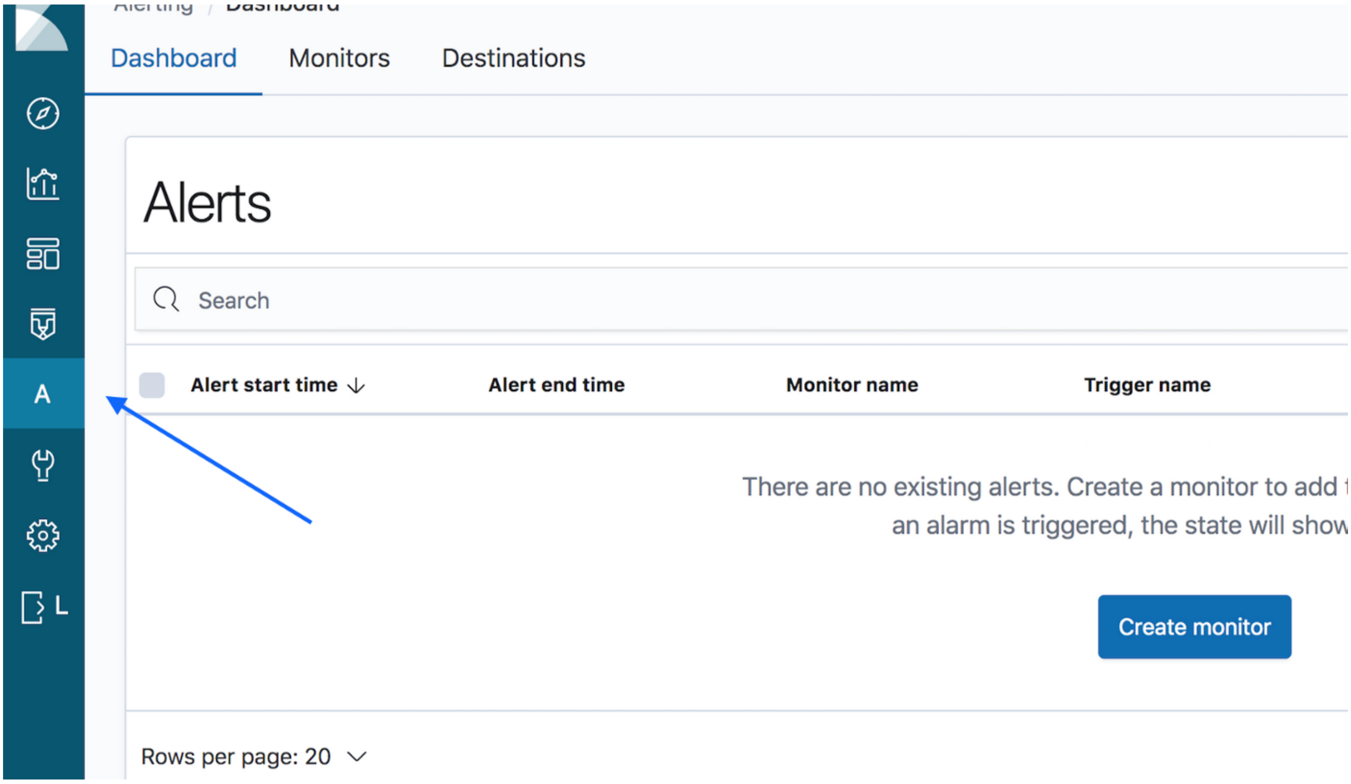
Craig Godden-Payne
Feb 20 · 3 min read ★

AWS Managed Elasticsearch now has functionality to monitor and alert from Kibana. In the past I have achieved similar functionality using Elastalert, but it has proved to be a bit clunky, so achieving similar functionality within Kibana if preferable to what I need to achieve.

The current setup where I am consulting is to use Grafana, to monitor for changes in logs, which works well due to easy setup of notifications, and ability to point to many different data sources. The problem with Grafana is that within notification channels, it is not possible to use variables, which means you have to identify every single possible problem that may happen, and write some static message in order to send relevant information when the alert is sent out, otherwise you have to stick to generic messages, and hope the person receiving the alert knows where to look.

Alerting is found on the left menu





Alerts

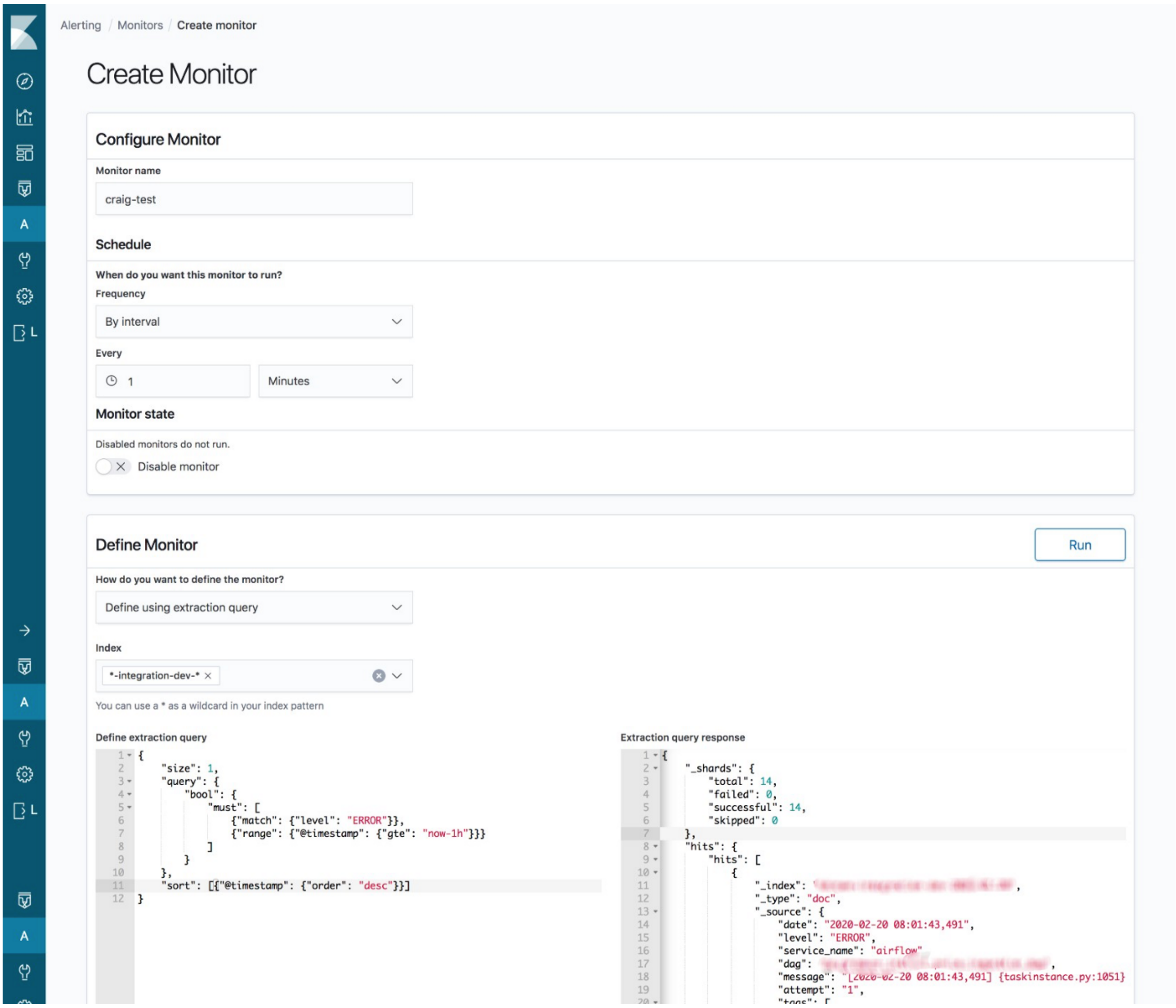
Search

☐	Alert start time ↓	Alert end time	Monitor name	Trigger name

There are no existing alerts. Create a monitor to add t
an alarm is triggered, the state will show

Rows per page: 20

Once you open the alerting section, the first thing to do is create a monitor.



Create Monitor

Configure Monitor

Monitor name: craig-test

Schedule

When do you want this monitor to run?

Frequency: By interval

Every: 1 Minutes

Monitor state

Disabled monitors do not run.

○ Disable monitor

Define Monitor

How do you want to define the monitor?: Define using extraction query

Index: *-integration-dev-*

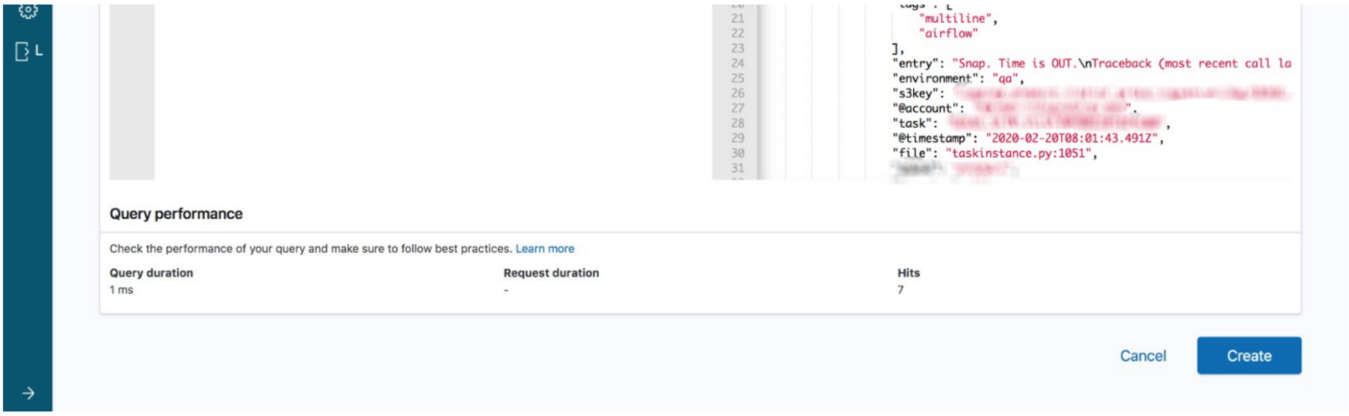
You can use a * as a wildcard in your index pattern

Define extraction query

```
{
    "size": 1,
    "query": {
        "bool": {
            "must": [
                {"match": {"level": "ERROR"}},
                {"range": {"@timestamp": {"gte": "now-1h"}}}
            ]
        }
    },
    "sort": [{"@timestamp": {"order": "desc"}}]
}
```

Extraction query response

```
{
    "_shards": {
        "total": 14,
        "failed": 0,
        "successful": 14,
        "skipped": 0
    },
    "hits": {
        "hits": [
            {
                "_index": [redacted],
                "_type": "doc",
                "_source": {
                    "date": "2020-02-20 08:01:43,491",
                    "level": "ERROR",
                    "service_name": "airflow"
                    "dag": [redacted],
                    "message": "[2020-02-20 08:01:43,491] {taskinstance.py:1051}
                    "attempt": "1",
                    "tags": [
```



```
    "multiline",
    "airflow"
],
"entry": "Snap. Time is OUT.\nTraceback (most recent call la
"environment": "qa",
"s3key": [redacted]
"@account": [redacted]",
"task": [redacted],
"@timestamp": "2020-02-20T08:01:43.491Z",
"file": "taskinstance.py:1051",
[redacted]
```

Query performance

Check the performance of your query and make sure to follow best practices. Learn more

Query duration 1 ms

Request duration -

Hits 7

I wrote a query which would find documents where the propery `level` is set to `ERROR` within the last hour, I also collect the first record, sorting by my timestamp.

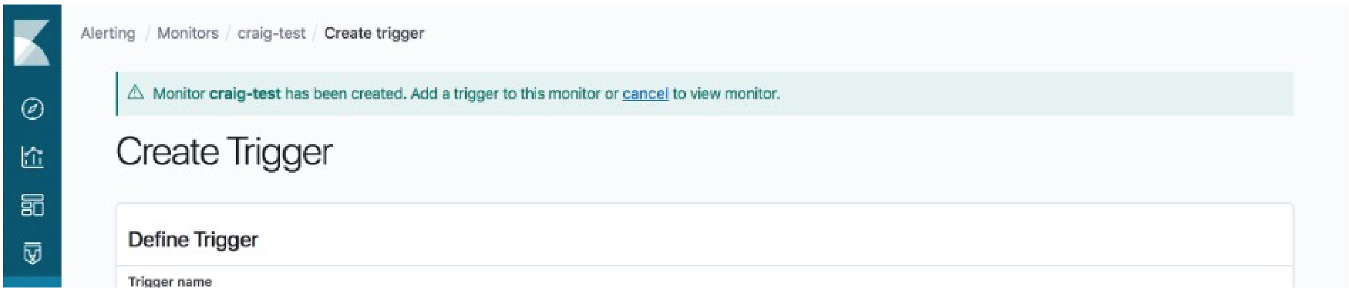
```
{
    "size": 1,
    "query": {
        "bool": {
            "must": [
                {"match": {"level": "ERROR"}},
                {"range": {"@timestamp": {"gte": "now-1h"}}}
            ]
        }
    },
    "sort": [{"@timestamp": {"order": "desc"}}]
}
```

I find using the dev tools in Kibana a great help, as it provides intellisense and autocomplete, when writing queries.

Once the monitor has been created, it is time to create the trigger.

I want to trigger an alert, whenever the count of hits is greater than 0, so I set my trigger condition to be:

```
ctx.results[0].hits.total > 0
```

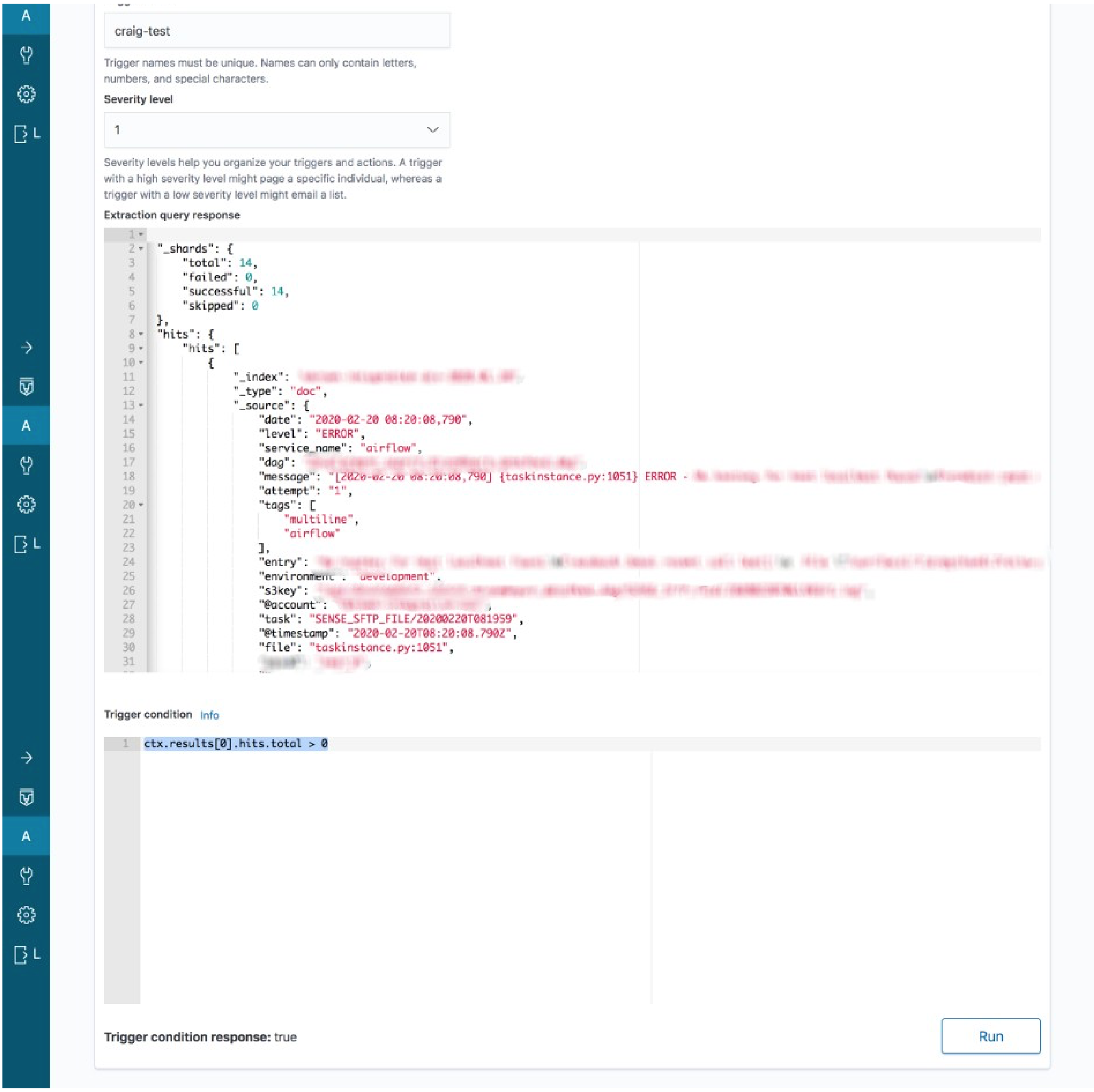


⚠ Monitor **craig-test** has been created. Add a trigger to this monitor or cancel to view monitor.

Create Trigger

Define Trigger

Trigger name



craig-test

Trigger names must be unique. Names can only contain letters, numbers, and special characters.

Severity level

1

Severity levels help you organize your triggers and actions. A trigger with a high severity level might page a specific individual, whereas a trigger with a low severity level might email a list.

Extraction query response

```

"_shards": {
    "total": 14,
    "failed": 0,
    "successful": 14,
    "skipped": 0
},
"hits": {
    "hits": [
        {
            "_index": [redacted]
            "_type": "doc",
            "_source": {
                "date": "2020-02-20 08:20:08,790",
                "level": "ERROR",
                "service_name": "airflow",
                "dag": [redacted]
                "message": "[2020-02-20 08:20:08,790] {taskinstance.py:1051} ERROR - [redacted]
                "attempt": "1",
                "tags": [
                    "multiline",
                    "airflow"
                ],
                "entry": [redacted]
                "environment": "development",
                "s3key": [redacted]
                "@account": [redacted]
                "task": "SENSE_SFTP_FILE/20200220T081959",
                "@timestamp": "2020-02-20T08:20:08.790Z",
                "file": "taskinstance.py:1051",
                [redacted]
```

Trigger condition Info

```
ctx.results[0].hits.total > 0
```

Trigger condition response: true

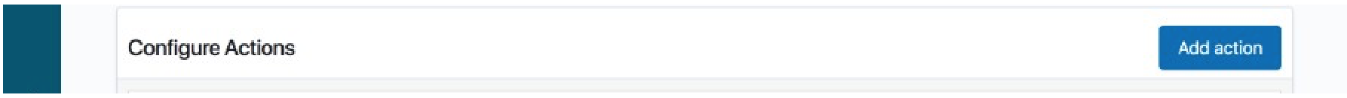
Next is configuring the action.

This was the part I was most interested in, as I want to be able to send the found message via the notification channel, something which I could not do with Grafana.

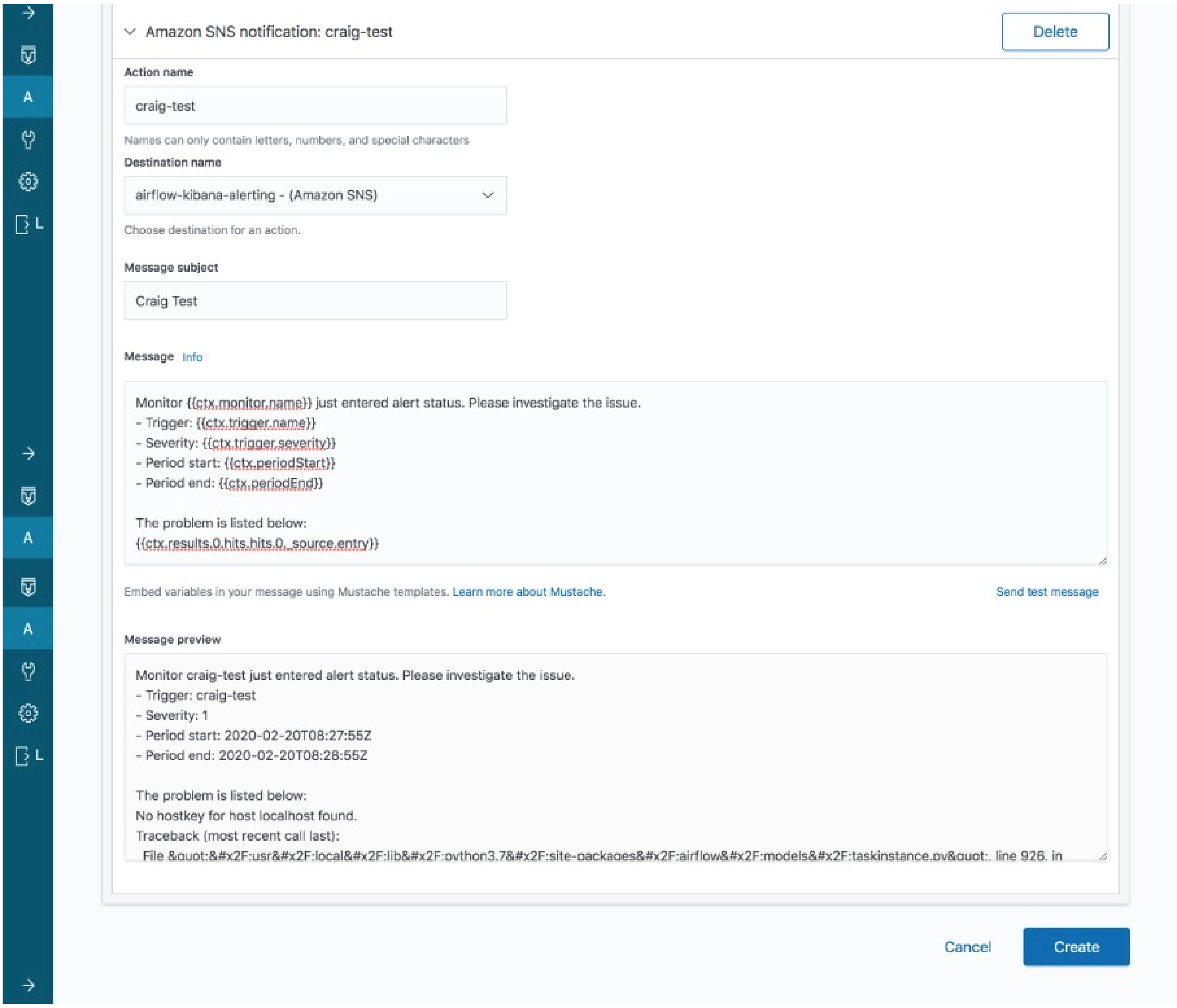
The notification message gives the following message:

```
 You have access to a "ctx" variable in your painless scripts and action mustache templates.
```

Since I want to send the ‘entry’ value in my notification, I can achieve this like so:



Configure Actions



Action name

craig-test

Names can only contain letters, numbers, and special characters

Destination name

airflow-kibana-alerting - (Amazon SNS)

Choose destination for an action.

Message subject

Craig Test

Message Info

```
Monitor {{ctx.monitor.name}} just entered alert status. Please investigate the issue.
- Trigger: {{ctx.trigger.name}}
- Severity: {{ctx.trigger.severity}}
- Period start: {{ctx.periodStart}}
- Period end: {{ctx.periodEnd}}

The problem is listed below:
{{ctx.results.0.hits.hits.0._source.entry}}
```

Embed variables in your message using Mustache templates. Learn more about Mustache.

Message preview

```
Monitor craig-test just entered alert status. Please investigate the issue.
- Trigger: craig-test
- Severity: 1
- Period start: 2020-02-20T08:27:55Z
- Period end: 2020-02-20T08:28:55Z

The problem is listed below:
No hostkey for host localhost found.
Traceback (most recent call last):
  File &quot;&#x2F;usr&#x2F;local&#x2F;lib&#x2F;python3.7&#x2F;site-packages&#x2F;airflow&#x2F;models&#x2F;taskinstance.py&quot;, line 926, in
```

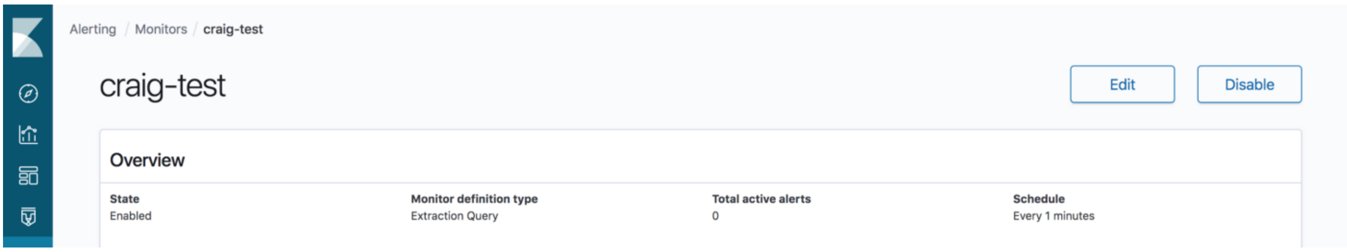
I found not much documentation about painless, so had a few issues accessing array variables, but I quickly figured out you could do this like so:

```
{{ctx.results.0.hits.hits.0._source.entry}}
```

Now everything is setup, I wanted to check everything worked as expected.

I actually wired my alert to an SNS topic, and just subscribed my email to all events.

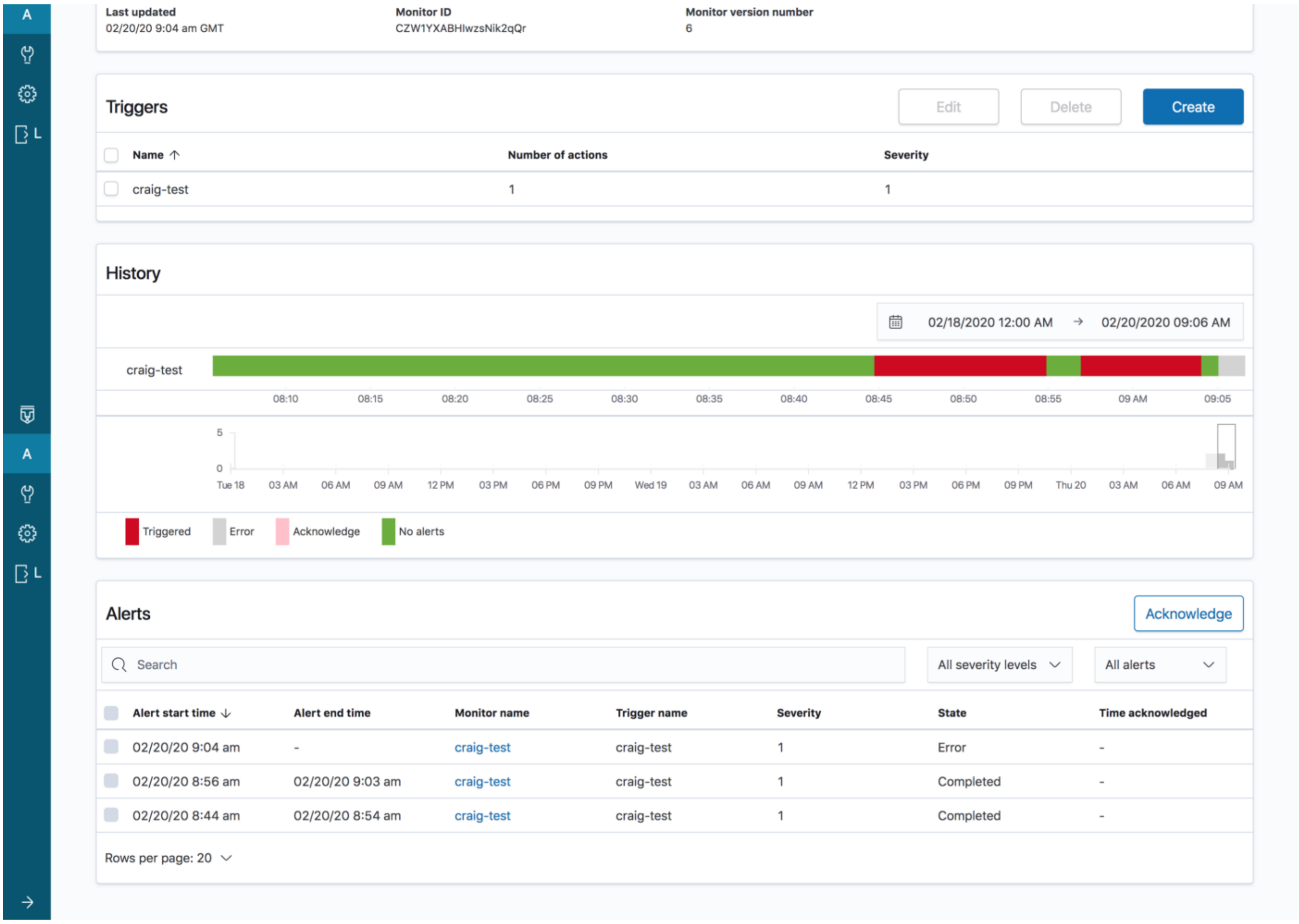
Here it is in action:



craig-test

Overview

State	Monitor definition type	Total active alerts	Schedule
Enabled	Extraction Query	0	Every 1 minutes



Last updated	Monitor ID	Monitor version number
02/20/20 9:04 am GMT	CZW1YXABHlwzsNik2qQr	6

Triggers

□	Name ↑	Number of actions	Severity
□	craig-test	1	1

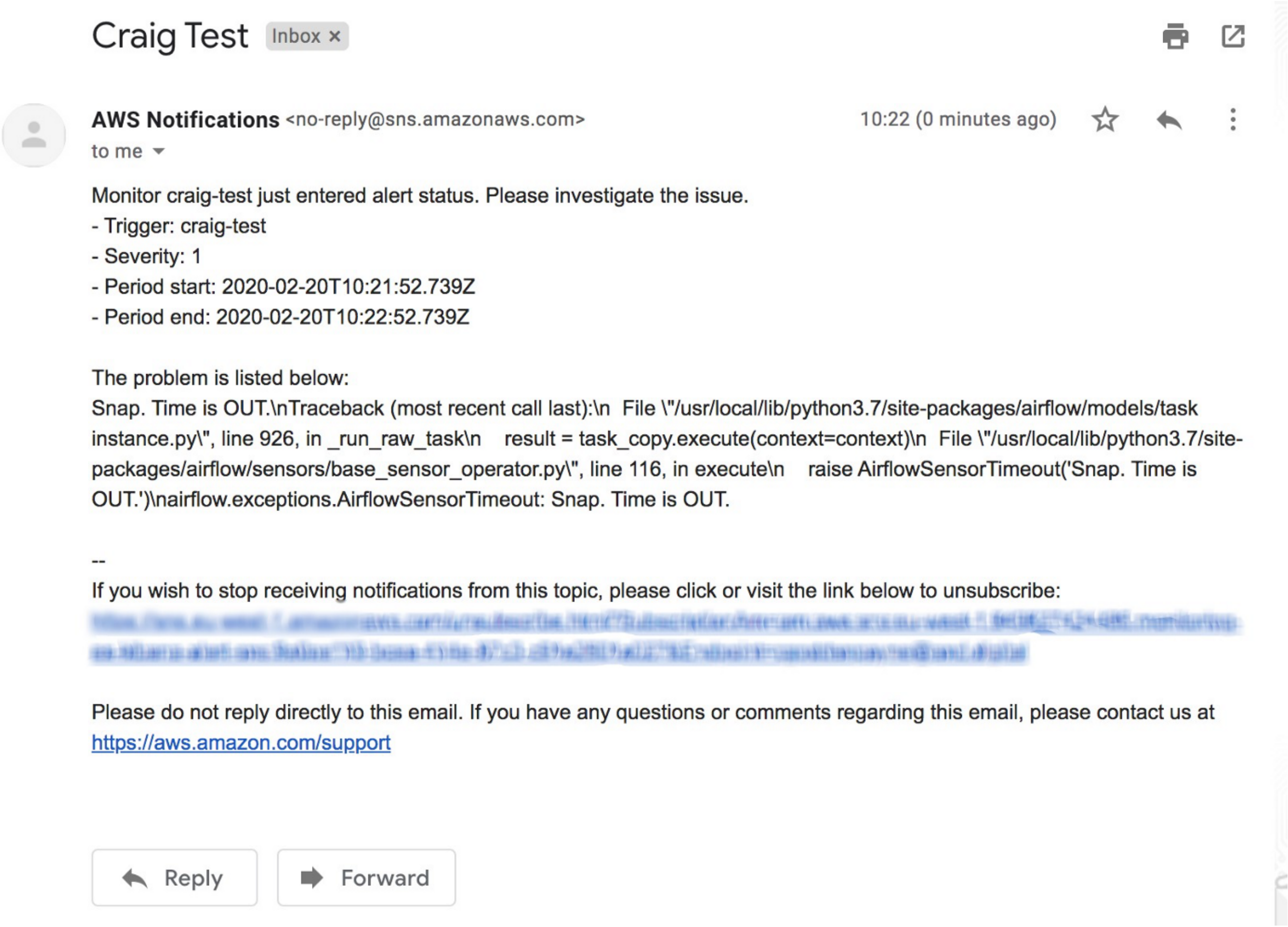
History

Alerts

□	Alert start time ↓	Alert end time	Monitor name	Trigger name	Severity	State	Time acknowledged
□	02/20/20 9:04 am	-	craig-test	craig-test	1	Error	-
□	02/20/20 8:56 am	02/20/20 9:03 am	craig-test	craig-test	1	Completed	-
□	02/20/20 8:44 am	02/20/20 8:54 am	craig-test	craig-test	1	Completed	-

Rows per page: 20

And here was the alert:



Craig Test Inbox ×

AWS Notifications <no-reply@sns.amazonaws.com>
to me

10:22 (0 minutes ago)

Monitor craig-test just entered alert status. Please investigate the issue.
- Trigger: craig-test
- Severity: 1
- Period start: 2020-02-20T10:21:52.739Z
- Period end: 2020-02-20T10:22:52.739Z

The problem is listed below:
Snap. Time is OUT.\nTraceback (most recent call last):\n File \"/usr/local/lib/python3.7/site-packages/airflow/models/taskinstance.py\", line 926, in _run_raw_task\n   result = task_copy.execute(context=context)\n File \"/usr/local/lib/python3.7/site-packages/airflow/sensors/base_sensor_operator.py\", line 116, in execute\n   raise AirflowSensorTimeout('Snap. Time is OUT.')\nairflow.exceptions.AirflowSensorTimeout: Snap. Time is OUT.

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
[redacted]

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support