Criar alerta para APM - Elasticsearch com ElastAlert 2

Descrição: Como criar um alerta para apm/elasticsearch utilizando o framework ElastAlert 2

Proprietário: Time NOC

Versão 1.0

Atualizado em: 15/02/2024

Introdução

Para a criação de um alerta no elasticsearch/opensearch, vamos utilizar o framework chamado ElastAlert, que é uma ferramenta bem poderosa, onde podemos criar diversos alertas através de metadados.

Sobre o ElastAlert: “O ElastAlert 2 é um framework simples para alertar sobre anomalias, picos ou outros padrões de interesse a partir de dados no Elasticsearch e no OpenSearch .”

1º Passo:

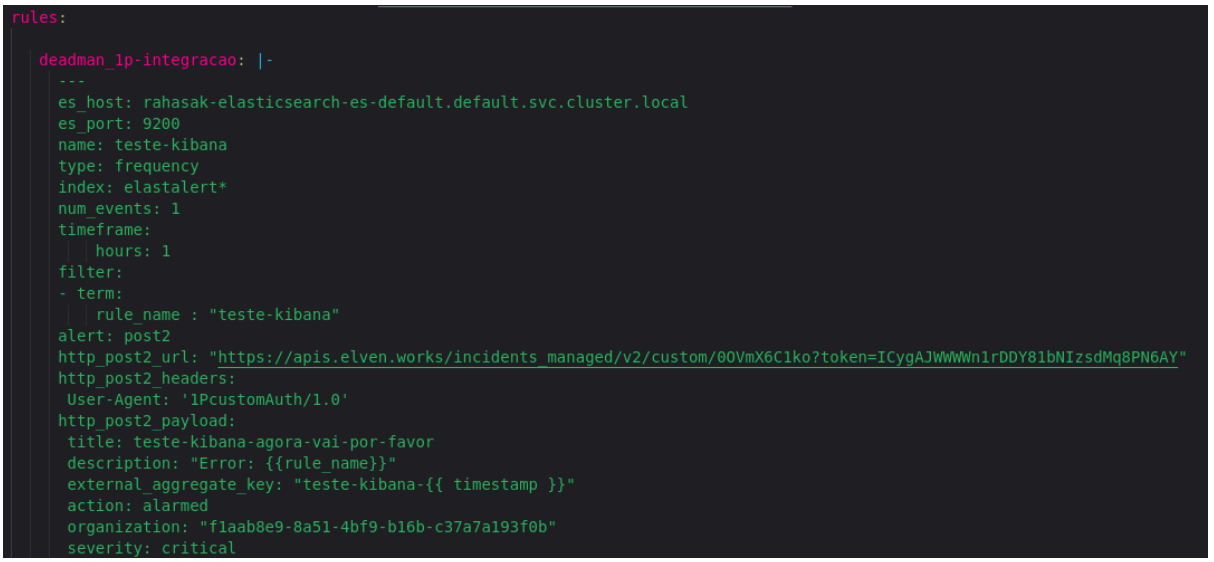
- Precisamos instalar o framework no seu ambiente e neste exemplo de instalação, será de acordo com a instalação em um k8s, através do Helm, onde o mesmo se encontra através do github oficial do framework, link aqui.

2º Passo:

- Precisamos fazer algumas alterações no arquivo de values do elastalert, como, incluir os dados para conexão com o elasticsearch, segue exemplo na imagem abaixo:

```yaml
elasticsearch:
  # elasticsearch endpoint e.g. (svc.namespace||svc)
  host: elasticsearch-es-default.default.svc.cluster.local
  # elasticsearch port
  port: 9200
  # whether or not to connect to es_host using TLS
  useSsl: "True"
  # Username if authenticating to ES with basic auth
  username: "elastic"
  # Password if authenticating to ES with basic auth
  password: "31y6I9wN9FW00azU4dzY5s4X"
  # Specifies an existing secret to be used for the ES username/password
  credentialsSecret: ""
  # The key in elasticsearch.credentialsSecret that stores the ES password
  credentialsSecretUsernameKey: ""
  # The key in elasticsearch.credentialsSecret that stores the ES username
  credentialsSecretPasswordKey: ""
  # whether or not to verify TLS certificates
  verifyCerts: "False"
```

E também, inserir em **rules** as configurações para os alertas que desejem, segue exemplo abaixo:



```
rules:

  deadman_1p-integracao: |-
    ---
    es_host: rahasak-elasticsearch-es-default.default.svc.cluster.local
    es_port: 9200
    name: teste-kibana
    type: frequency
    index: elastalert*
    num_events: 1
    timeframe:
        hours: 1
    filter:
    - term:
        rule_name : "teste-kibana"
    alert: post2
    http_post2_url: "https://apis.elven.works/incidents_managed/v2/custom/0OVmX6C1ko?token=ICygAJWWWWn1rDDY81bNIzsdMq8PN6AY"
    http_post2_headers:
     User-Agent: '1PcustomAuth/1.0'
    http_post2_payload:
     title: teste-kibana-agora-vai-por-favor
     description: "Error: {{rule name}}"
     external_aggregate_key: "teste-kibana-{{ timestamp }}"
     action: alarmed
     organization: "f1aab8e9-8a51-4bf9-b16b-c37a7a193f0b"
     severity: critical
```

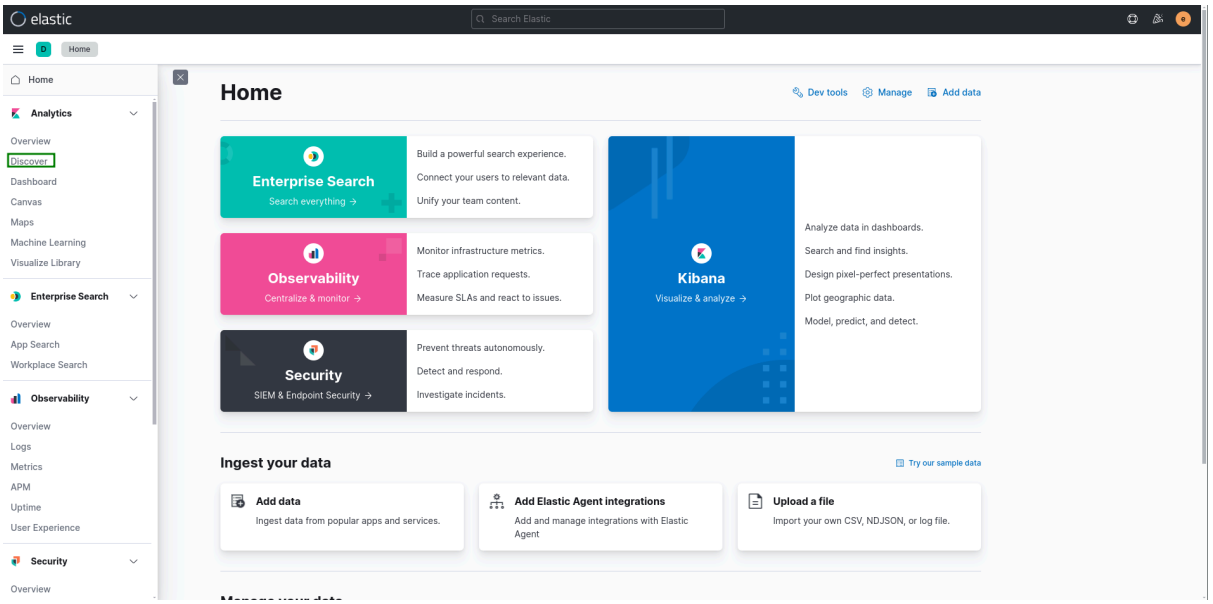
No exemplo acima, é feito a configuração de alerta, através de uma custom integration com a 1P, onde o alerta irá disparar como alarmed, se dentro do index de nome **elastalert***, ter uma **rule_name** igual a "**teste_kibana**", irá gerar um alerta e assim a integração vai fazer gerar um alerta na 1P.

ElastAlert

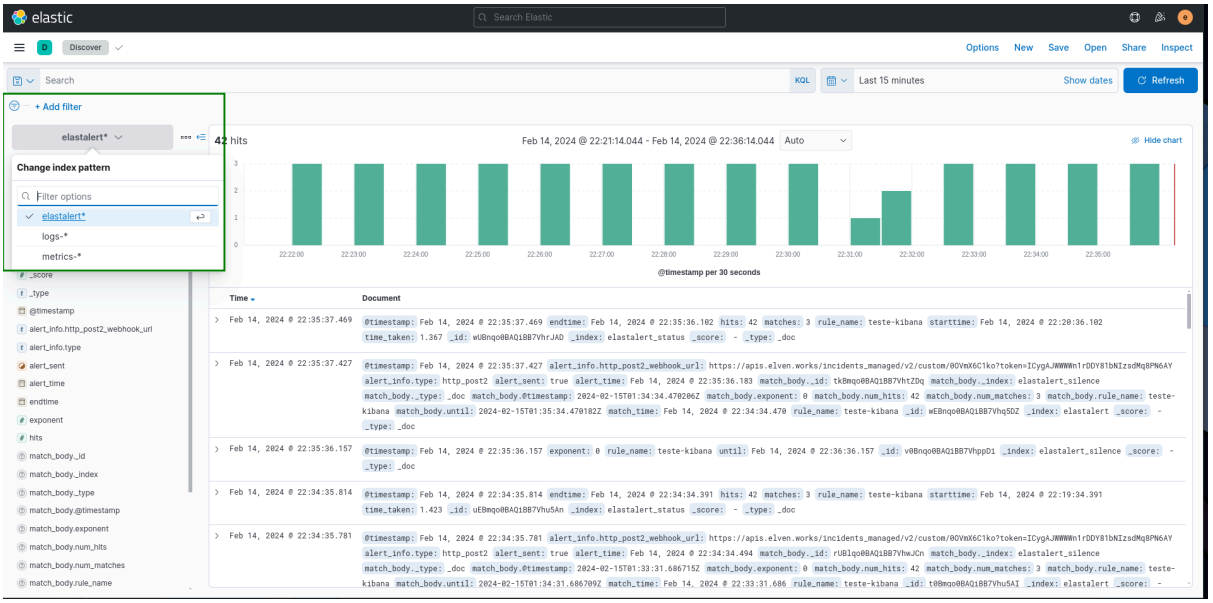
Funciona combinando o Elasticsearch com dois tipos de componentes, regras e alertas. O datasource é consultado periodicamente e os dados são passados através de uma rule type, que determina quando uma ocorrência é encontrada, quando ocorre um match entre a rule e o datasource, é disparado um ou mais alertas, nos quais a ação é baseado no tipo de match.

Existem vários tipos de regras que podem ser configuradas de diversas maneiras, no exemplo, vou detalhar cada uma de acordo com o exemplo acima:
es_host: Service dentro do cluster que irá fazer a conexão entre o framework e o elasticsearch
es_port: Porta que será realizada esta conexão
name: Nome do alerta que será exibido
type: Tipo de alerta que será criado (existem diversos tipos de alertas, veja aqui.)
Obs: O tipo de regra **frequency**, corresponde quando a pelo menos um certo número de eventos em um determinado período de tempo, estas informações podem ser configuradas também.

index: O index que será consultado, pode ser uma lista de index (essa informação pode ser visualizada no elasticsearch, na tela de **discover**)



E é possível visualizar quais os index patterns disponíveis



num_events: Número de ocorrências para alarmar
timeframe: O período a ser pesquisado, podendo ser horas ou minutos
filter: existem vários tipos e podem incluir uma lista deles, existem filtros que fazem a consulta por filtros de string, termo exato, mais de um termo, etc, segue a documentação com vários exemplos dos tipos de filtros aqui.
alert: Para onde será enviado o alerta, pode-se enviar os alertas para diversas aplicações, e o tipo http2 envia os resultados para um endpoint JSON usando o método HTTP POST, onde o nome e as chaves são configuráveis, assim, sendo compatíveis com quase todos os endpo