



# EL(B)K

Elasticsearch  
Logstash/Beats  
Kibana



Kibana



Elasticsearch



Beats

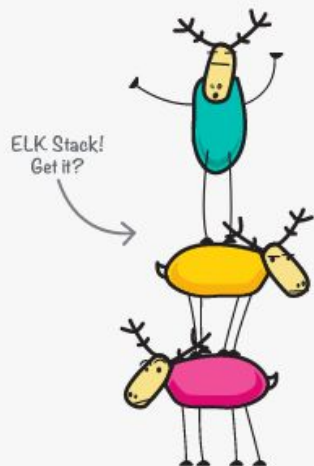


Logstash

## Tudo começou com o Elasticsearch...

O mecanismo de busca open source, distribuído, RESTful e baseado em JSON. Fácil de usar, escalável e flexível, ele conquistou uma superpopularidade entre os usuários e uma empresa se formou em torno dele para fazer buscas.





**E** Elasticsearch

**L** Logstash

**K** Kibana

## E cresceu com o Logstash e o Kibana

O Elasticsearch é um mecanismo de busca em seu núcleo, e os usuários começaram a usá-lo para logs e queriam que fosse possível fazer a ingestão e visualização deles facilmente. Daí entram o Logstash, o avançado pipeline de ingestão, e o Kibana, a ferramenta flexível de visualização.



## A comunidade ficou maior, e os casos de uso, mais numerosos

Fosse para encontrar os principais N resultados em um monte de documentos de texto, para analisar eventos de segurança ou para segmentar metrics livremente, a comunidade mundial continuou extrapolando os limites com o ELK.





## Daí implantamos um Beat no ELK

Os usuários diziam "Eu só quero fazer tailing em um arquivo". E nós ouvimos. Em 2015, apresentamos uma família de agentes de dados leves de finalidade única na equação do ELK Stack. Demos a eles o nome Beats.

O E do ELK:  
Elasticsearch






## Principais Características:

- Banco de dados orientado a documento
- Engine de busca
- Rápido
- Escalável
- API Rest para ingestão, busca de dados, etc...
- Open source





Clientes oficiais em diversas linguagens:

- Java
- Javascript/Node.js
- Go
- .NET (c#)
- PHP
- Perl
- Python
- Ruby

# Para que o Elasticsearch é usado?

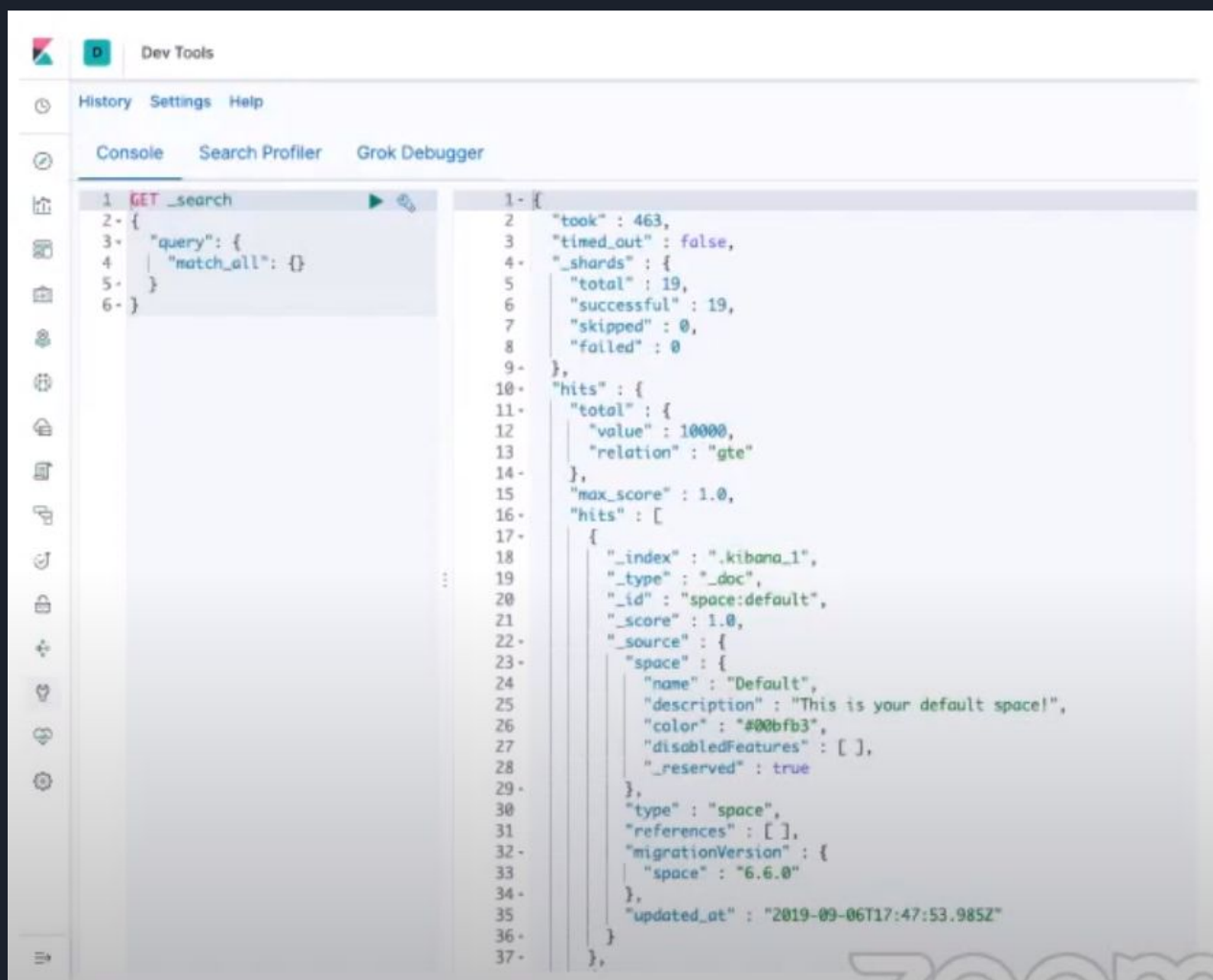
A velocidade e escalabilidade do Elasticsearch e sua capacidade de indexar muitos tipos de conteúdo significam que ele pode ser usado para inúmeros casos de uso:

- Busca em aplicação
- Busca em website
- Busca empresarial
- Logging e análise de dados de log
- Metrics de infraestrutura e monitoramento de containers
- APM (Monitoramento de performance de aplicação)
- Análise e visualização de dados geoespaciais
- Análise de segurança
- Análise de dados empresarial

Exemplo de query no banco de dados do elasticsearch

Depois que os dados são indexados no Elasticsearch, os usuários podem executar consultas complexas com base em seus dados e usar agregações para recuperar resumos complexos dos dados.

Devido a poderosa indexação do elasticsearch, consultas em bases de dados gigantes são altamente otimizadas.



The screenshot shows the DevTools console with the following content:

```
History Settings Help
Console Search Profiler Grok Debugger

1 GET _search
2 {
3   "query": {
4     "match_all": {}
5   }
6 }

1- {
2   "took": 463,
3   "timed_out": false,
4   "_shards": {
5     "total": 19,
6     "successful": 19,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 10000,
13      "relation": "gte"
14    },
15    "max_score": 1.0,
16    "hits": [
17      {
18        "_index": ".kibana_1",
19        "_type": "_doc",
20        "_id": "space:default",
21        "_score": 1.0,
22        "_source": {
23          "space": {
24            "name": "Default",
25            "description": "This is your default space!",
26            "color": "#00bfb3",
27            "disabledFeatures": [ ],
28            "_reserved": true
29          },
30          "type": "space",
31          "references": [ ],
32          "migrationVersion": {
33            "space": "6.6.0"
34          },
35          "updated_at": "2019-09-06T17:47:53.985Z"
36        }
37      },
38    ]
39  }
40 }
```

# Elasticsearch vs Mongo





# As semelhanças

## Elasticsearch

- Plug and play;
- Alta disponibilidade, capacidade, resiliência e escalabilidade
- schema-free
- sharding

## Mongo

- Plug and play
- Alta disponibilidade, capacidade, resiliência e escalabilidade
- schema-free
- sharding



# As diferenças

## Elasticsearch

O elasticsearch não possui nenhum suporte a transactions. Um DELETE em seu índice não verifica condições de integridade, seus dados são apenas deletados.

O elasticsearch por sua vez é mais apropriado para full-text search, Stemming, buscas geoespaciais, logs, classificação e ranking de resultados de pesquisa.

## Mongo

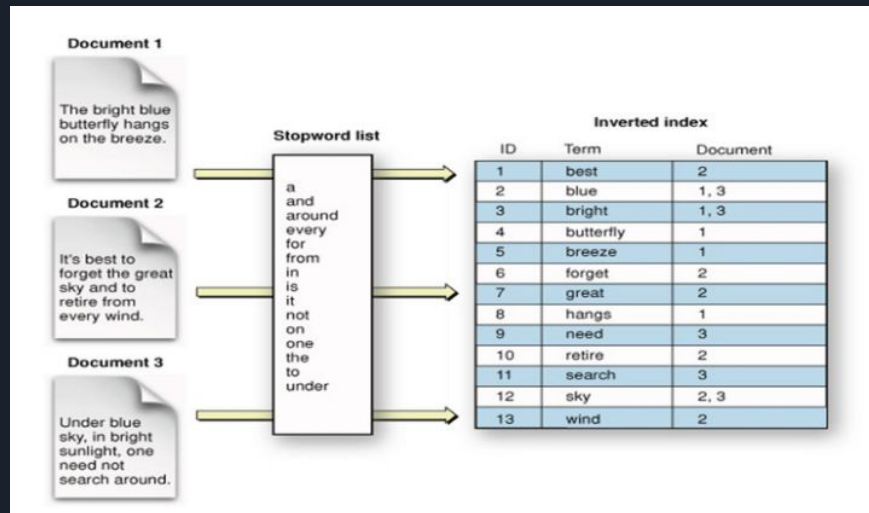
O mongo tem suporte a ACID (Atomicidade, Consistência, Isolamento e Durabilidade) transactions , tornando-o mais apropriado que elasticsearch para servir de banco de dados primário de uma aplicação.

# As diferenças

## Elasticsearch

O elasticsearch possui um sistema de indexamento em tempo real utilizando o inverted index do apache lucene, indexando cada campo de cada documento por padrão, otimizando-o para buscas em larga escala.

A indexação pode acabar levando mais tempo, mas suas pesquisas e agregações podem ser ordens de grandeza mais rápidas e complexas em comparação a sistemas semelhantes.





# Queries

## Elasticsearch

API\_search acessada via REST utilizando JSON como base.

## Mongo

Linguagem própria.





# O melhor?

Apesar de a princípio se parecerem, logo fica claro que servem propósitos distintos, sendo possível e até recomendado que, caso seja necessário, utilizar os dois em conjunto.

O mongo é um banco de dados não relacional genérico, sendo apropriado para armazenamento de dados.

O elasticsearch é excepcional para indexação, sendo recomendado para pesquisas e agregações de grandes volumes de dados como logs. Com foco nas últimas releases em de machine learning e analytics. Ainda assim é possível utilizar o elasticsearch como banco de dados como cita Alex Brasetvik:

"It is certainly possible to use Elasticsearch as a primary store, when the limitations described are not showstoppers." — BRASETVIK, Alex

<https://www.elastic.co/pt/blog/found-elasticsearch-as-nosql>

<https://medium.com/data-hackers/comparando-elasticsearch-vs-mongodb-4b5932c613d9>

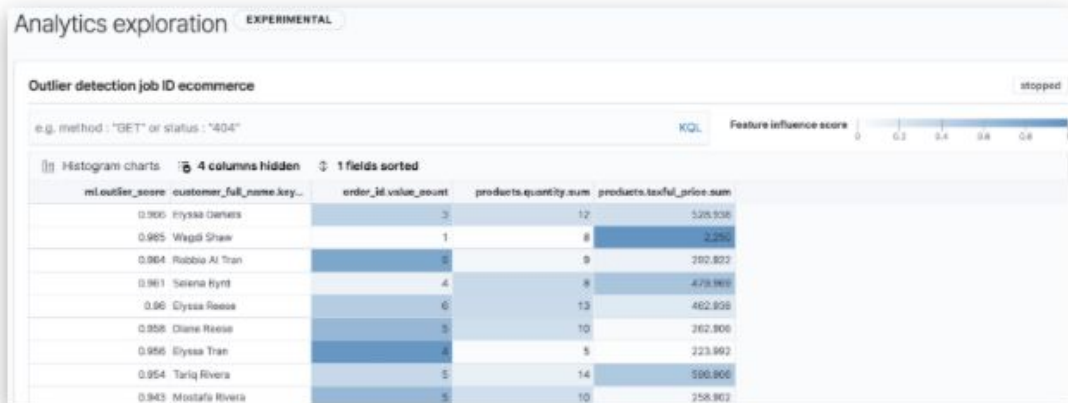
Plugins



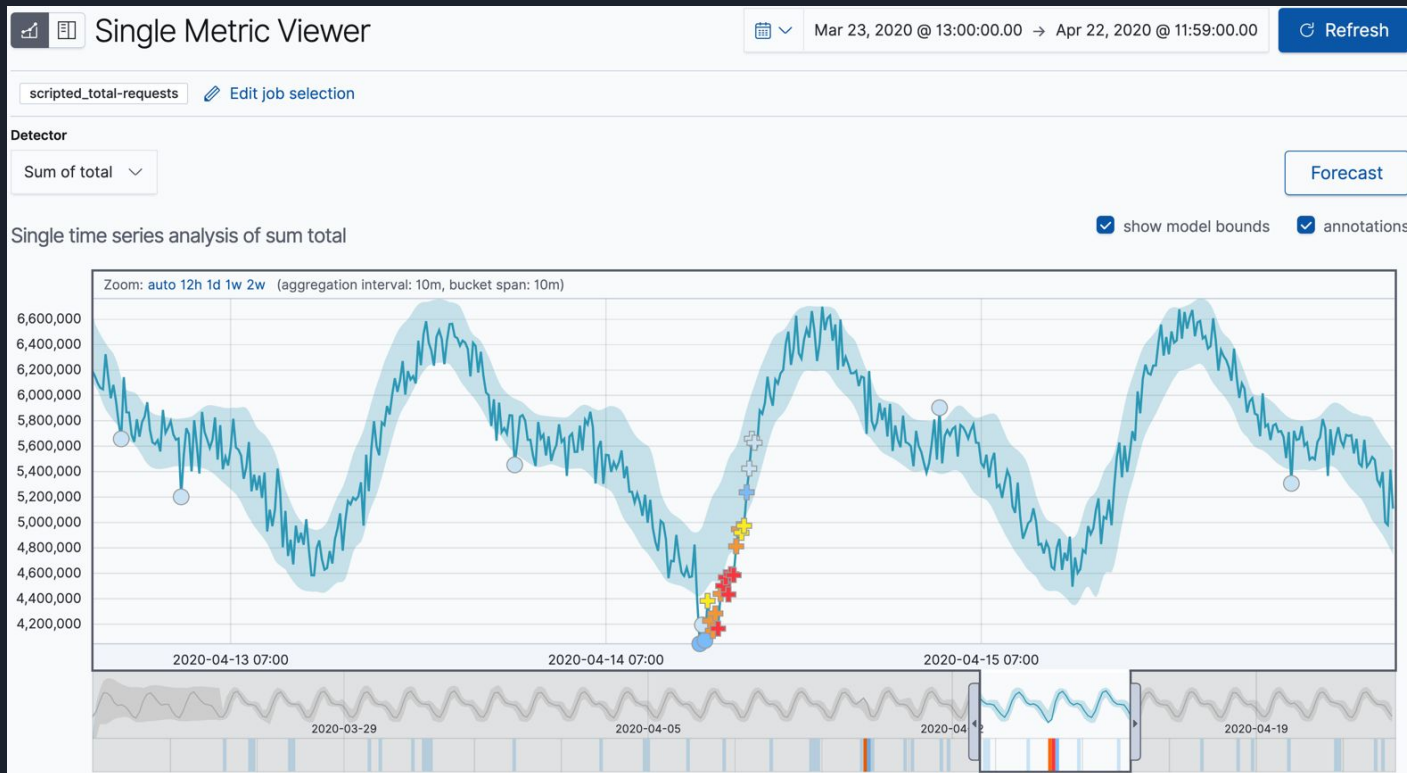
# Plugin Machine Learning - Adquirido via assinatura na elastic.co

The Elastic machine learning data frame analytics feature enables you to analyze your data using classification, outlier detection, and regression algorithms and generate new indices that contain the results alongside your source data.

If you have a license that includes the machine learning features, you can create data frame analytics jobs and view their results on the **Data Frame Analytics** page in Kibana. For example:



# Plugin Anomaly Detection - Adquirido via assinatura na elastic.co





# Plugin Open Distro - Open Source

- Log analytics
- Real-time application monitoring
- Clickstream analytics
- Search backend
- Anomaly detection
- KNN query



# Outros plugins Open Source

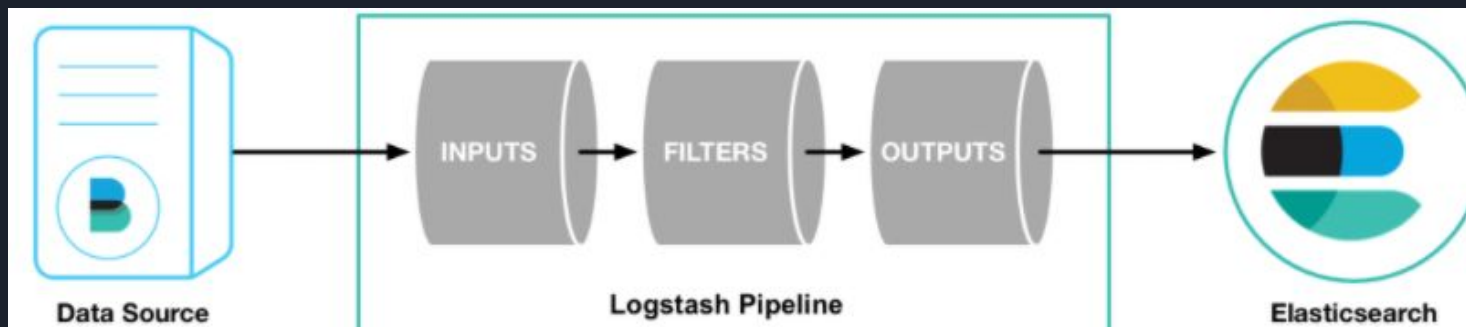
- Elasticsearch Learning To Rank ★1,101 - Plugin to integrate Learning to Rank (aka machine learning for better relevance) with Elasticsearch
- Gem ★655 - GUI for Data Modeling with Elasticsearch
- Alerting ★236 - Open Distro for Elasticsearch Alerting Plugin
- Graph Aided Search ★148 - Elasticsearch plugin offering Neo4j integration for Personalized Search
- Elastiknn ★106 - Elasticsearch plugin for nearest neighbor search. Store vectors and run similarity search using exact and approximate algorithms.

O L do ELK:  
Logstash



### Principais características:

- Faz a coleta de dados em múltiplas fontes e envia para múltiplas fontes (data shipper)
- Pré processamento dos dados aplicando filtros e transformadores
- Envia para o dado para seu destino final, podendo ser para o elasticsearch, ou até mesmo para um arquivo no disco local.
- Open source








# Exemplos de configuração do logstash

```
input { stdin { } }

filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
  date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
}

output {
  elasticsearch { hosts => ["localhost:9200"] }
  stdout { codec => rubydebug }
}
```



```
127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] "GET /xampp/status.php HTTP/1.1" 200 3891
"http://cadenza/xampp/navi.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101
Firefox/25.0"
```

```
{
  "message" => "127.0.0.1 - - [11/Dec/2013:00:01:45 -0800] \"GET /xampp/sta
"@timestamp" => "2013-12-11T08:01:45.000Z",
  "@version" => "1",
  "host" => "cadenza",
  "clientip" => "127.0.0.1",
  "ident" => "-",
  "auth" => "-",
  "timestamp" => "11/Dec/2013:00:01:45 -0800",
  "verb" => "GET",
  "request" => "/xampp/status.php",
  "httpversion" => "1.1",
  "response" => "200",
  "bytes" => "3891",
  "referrer" => "\"http://cadenza/xampp/navi.php\"",
  "agent" => "\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Geo
}
```

# Exemplos de padrões (GROK) para transformação de dados no pipeline do logstash

## Sample Data

```
1 83.149.9.216 - - [17/May/2015:10:05:03 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1" 200 203023 "http://semicon
```

## Grok Pattern

```
1 %{IP:ip}
```

> Custom Patterns

Simulate

## Structured Data

```
1 {  
2   "ip": "83.149.9.216"  
3 }
```



Grok patterns prontos:

<https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/grok-patterns>

```

USERNAME [a-zA-Z0-9._-]+
USER %{USERNAME}
EMAILLOCALPART [a-zA-Z][a-zA-Z0-9_+-.:~]+
EMAILADDRESS %{EMAILLOCALPART}@%{HOSTNAME}
INT (?:[+-]?(?:[0-9]+))
BASE10NUM (?<![0-9.-+<?>[+-]?(?:(?:[0-9]+(?:(?!\.[0-9])?))|(?!\.[0-9])?))
NUMBER (?:%{BASE10NUM})
BASE16NUM (?<![0-9A-Fa-f])(?:[+-]?(?:(?![0-9A-Fa-f])|(?:[0-9A-Fa-f]))+)
BASE16FLOAT \b(?<![0-9A-Fa-f.])?(?:[+-]?(?:(?![0-9A-Fa-f])|(?:[0-9A-Fa-f]+(?:(?!\.[0-9A-Fa-f])?))|(?!\.[0-9A-Fa-f])?))\b

POSINT \b(?:[1-9][0-9]*)\b
NONNEGINT \b(?:[0-9]+)\b
WORD \b\w+\b
NOTSPACE \S+
SPACE \s*
DATA .*?
GREEDYDATA .*

QUOTEDSTRING (?>(?!\\)(?>"(?>\\.|[^\\""]+)"|'(?>\\.|[^\\"']+)'|'(?>\\.|[^\\"']+)'|''))
UUID [A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}
# URN, allowing use of RFC 2141 section 2.3 reserved characters
URN urn:[0-9A-Za-z][0-9A-Za-z-]{0,31}:(?:%[0-9a-fA-F]{2}|[0-9A-Za-z()+,.:=@$_!*/?#-])+

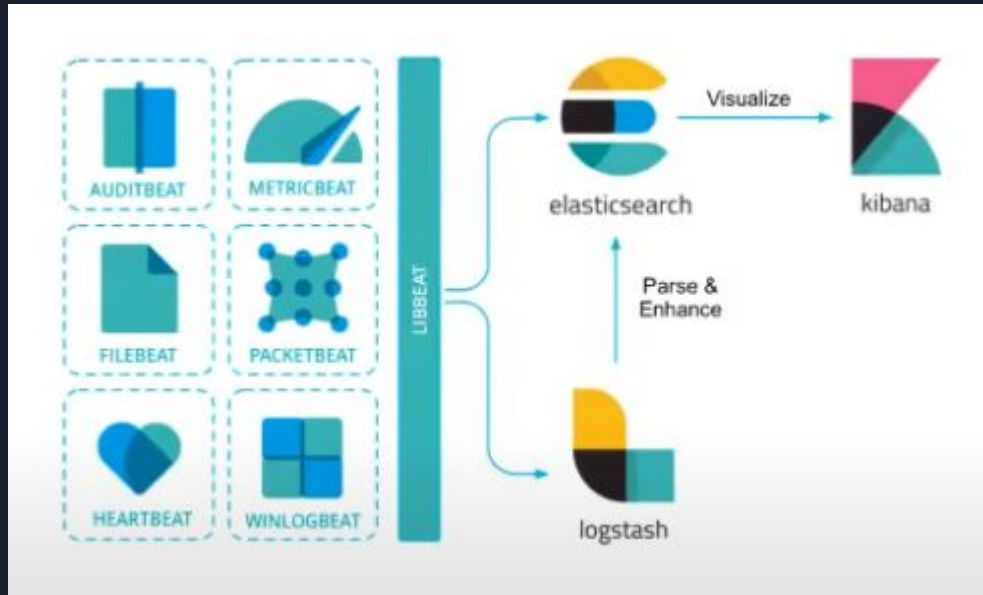
# Networking
MAC (?:%{CISCOMAC}|%{WINDOWSMAC}|%{COMMONMAC})
CISCOMAC (?:(?:[A-Fa-f0-9]{4}\.){2}[A-Fa-f0-9]{4})
WINDOWSMAC (?:(?:[A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2})
COMMONMAC (?:(?:[A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2})
IPV6 ((([0-9A-Fa-f]{1,4}:){7}([0-9A-Fa-f]{1,4}|:))|((([0-9A-Fa-f]{1,4}:){6}(:[0-9A-Fa-f]{1,4}|([25[0-5]|2[0-4]\d|1\d\d|1[0-9]?|\d)(\.(25[0-5]|2[0-4]\d|1\d\d|1[0-9]?|\d)){3})|:))|((([0-9A-Fa-f]{1,4}:){0,6})([0-9A-Fa-f]{1,4}|:)))|(:[0-9A-Fa-f]{1,4})|:))
IPV4 (?<![0-9])(?:(?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])|(?![0-9A-Fa-f])(?:(?![0-9A-Fa-f])|(?:[0-9A-Fa-f]))+)
IP (?:%{IPV6}|%{IPV4})
HOSTNAME \b(?:[0-9A-Za-z][0-9A-Za-z-]{0,62})(?!\.)(?:[0-9A-Za-z][0-9A-Za-z-]{0,62})*(\.|\b)
IPORHOST (?:%{IP}|%{HOSTNAME})
HOSTPORT %{IPORHOST}:%{POSINT}

# paths

```

# Beats

O beats é um “lightweight data shipper”, uma opção mais leve em comparação com o logstash, mas também pode ser utilizado em conjunto.



O K do ELK:  
Kibana





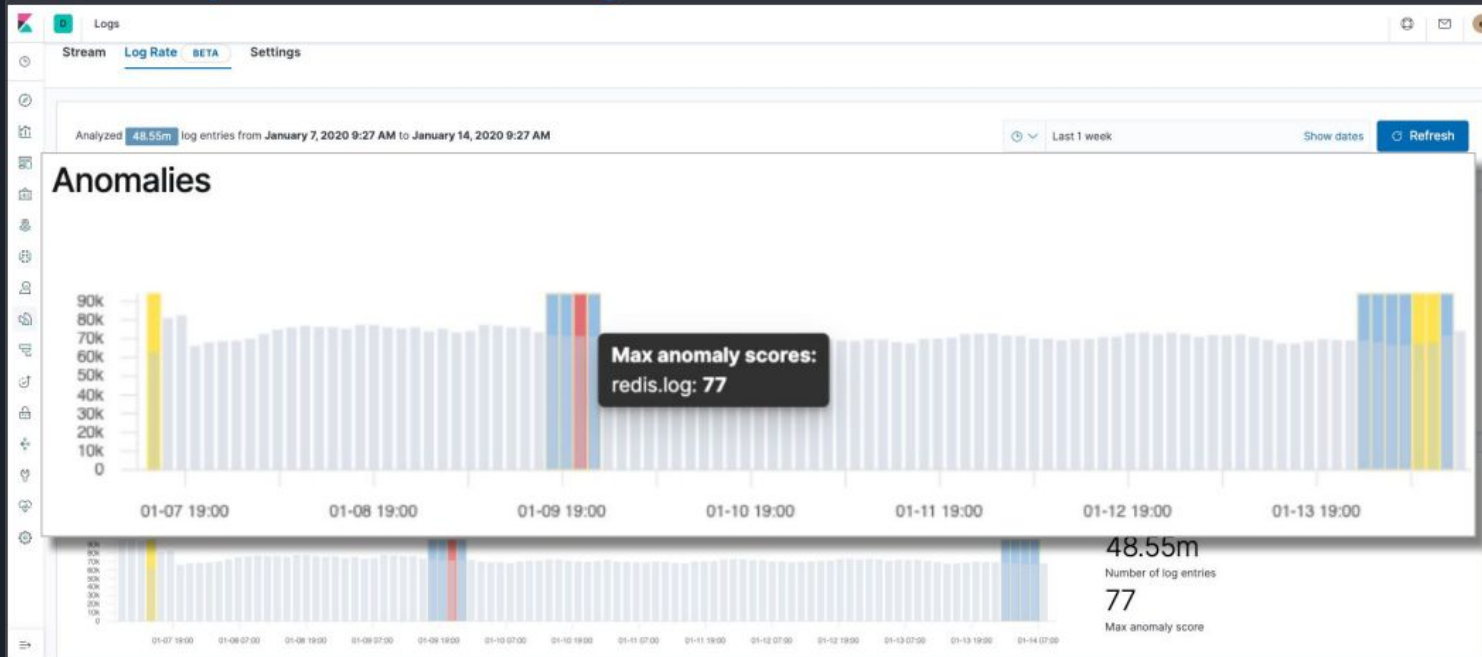
## Principais Características:

- Visualização e exploração dos dados
- Interface gráfica para consulta dos dados agregados no elasticsearch
- Possui ferramentas para próprias para criação de gráficos e dashboards utilizando os dados do elasticsearch
- Suporte a mapas
- Ferramentas de desenvolvedor. Ex: criação de padrões para formatação no logstash (grok pattern debugger)
- open source



# Needs for log analytics

## Anomaly detection and alerting



# Implementando no .NET Core

Para testar, crie um projeto novo no visual studio com o template de web api com .net core.  
Instale no seu projeto .NET Core:

1. Serilog;
2. Serilog.Sinks.ElasticSearch;
3. Serilog.Extensions.Logging;

Adicione ao appsettings.json:

```
1 {  
2  
3   "Logging": {  
4     "LogLevel": {  
5       "Default": "Information",  
6       "System": "Information",  
7       "Microsoft": "Information"  
8     }  
9   },  
10  "ElasticConfiguration": {  
11    "Uri": "http://elastic:changeme@localhost:9200/"  
12  },  
13  "AllowedHosts": "*"   
14 }  
15
```

## Em Startup.cs

0 references

```
public Startup(IConfiguration configuration)
{
    Configuration = configuration;

    var elasticUri = Configuration["ElasticConfiguration:Uri"];

    Log.Logger = new LoggerConfiguration()
        .Enrich.FromLogContext()
        .Enrich.WithMachineName()
        .WriteTo.Elasticsearch(new ElasticsearchSinkOptions(new Uri(elasticUri))
        {
            AutoRegisterTemplate = true,
            IndexFormat = $"logs-net-core-demo-{DateTime.UtcNow:yyyy-MM-dd}"
        })
        .CreateLogger();
}
```

## Em Startup.cs

```
U References
public void Configure(IApplicationBuilder app, IWebHostEnvironment env, ILoggerFactory loggerFactory)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }


    loggerFactory.AddSerilog();

    app.UseHttpsRedirection();

    app.UseRouting();

    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllers();
    });
}
```



```
[HttpGet]
```

```
0 references
```

```
public ActionResult Get()
```

```
{
```

```
    _logger.LogDebug("Este é um log de DEBUG");  
    _logger.LogInformation("Este é um log de INFO");  
    _logger.LogWarning("Este é um log de WARNING");  
    _logger.LogError("Este é um log de ERROR");  
    _logger.LogCritical("Este é um log de CRITICAL");
```

```
    return Ok(new { Sucess = true });
```

```
}
```

```
elk_netcore.Controllers.WeatherForecastController: Information: Este é um log de INFO  
elk_netcore.Controllers.WeatherForecastController: Warning: Este é um log de WARNING  
elk_netcore.Controllers.WeatherForecastController: Error: Este é um log de ERROR  
elk_netcore.Controllers.WeatherForecastController: Critical: Este é um log de CRITICAL
```



O código anterior envia os logs diretamente para o elasticsearch através do método

“WriteTo.ElasticSearch()”

É possível enviar para o logstash para que o mesmo faça o pré processamento e assim envie para o elasticsearch alterando o método “WriteTo” para:

```
var log = new LoggerConfiguration()  
    .WriteTo.Http("http://localhost:8080")  
    .CreateLogger();
```

E nas configurações do logstash:

```
input {  
  http {  
    #default host 0.0.0.0:8080  
    codec => json  
  }  
}  
  
output {  
  elasticsearch {  
    hosts => "elasticsearch:9200"  
    index=>"sales-%{+xxxx.wv}"  
  }  
}
```



# Implementando django


Instale as seguintes dependências:

```
pip install django-elasticsearch-dsl  
  
# Elasticsearch 5.x  
pip install 'elasticsearch-dsl>=5.0,<6.0'
```

# Em settings.py

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'formatters': {
        'simple': {
            'format': '[%(asctime)s] %(levelname)s|%(name)s|%(message)s',
            'datefmt': '%Y-%m-%d %H:%M:%S',
        },
    },
    'handlers': {
        'console': {
            'level': 'INFO',
            'class': 'logging.StreamHandler',
            'formatter': 'simple'
        },
        'logstash': {
            'level': 'DEBUG',
            'class': 'logstash.TCPLogstashHandler',
            'host': 'localhost',
            'port': 5000, # Default value: 5959
            'version': 1, # Version of logstash event schema. Default value: 0 (for backward compatibility of the library)
            'fqdn': False, # Fully qualified domain name. Default value: false.
        },
    },
    'loggers': {
        '': {
            'handlers': ['logstash', 'console'],
            'level': 'DEBUG',
            'propagate': True
        },
    },
}
```





## Em logstash.conf (arquivo de configuração do logstash)

```
input {  
  tcp {  
    port => 44342  
  }  
  tcp {  
    port => 5959  
    codec => json  
  }  
}  
  
output {  
  elasticsearch {  
    hosts => "localhost:9200"  
    user => "elastic"  
    password => "changeme"  
  }  
}
```



# Em alguma view:

```
class Test(APIView):  
    def get(self, request, *args, **kwargs):  
        logger.debug("debug log")  
        logger.info("info log")  
        logger.warning("warning log")  
        logger.error('error log')  
        logger.critical("critical log")  
        return Response({'sucess':True})
```