



Personalized Learning

Pós-Graduação em MTI Big Data

PBGAS15 – Gerencia Dados Bloco C –

Armazenamento Heterogêneo de Dados

Avaliação Módulo I – Bases NoSQL

- Elasticsearch -

Eduardo Bizarria Gaspar
Rafael Hermógenes Mendonça
Rodrigo Santos de Aquino
Tiago Ferreira da Silva Seabra
Pós-Graduandos

Eduardo Morelli
Coordenador / Orientador

Sumário

Objetivo	3
O Elasticsearch.....	4
Field Data Types	6
Requisitos e configurações.....	9
Mapeamento Dinâmico	11
Bulk.....	11
Cases	12
Indexação.....	13
Index Templates.....	14
Multiple Index Templates	15
Busca	17
Agregação	18
Time Based Data	19
Segurança	20
Logstash	21
Kibana	26



Objetivo

Esse documento visa mostrar o produto Elasticsearch, ou Elastic como é chamado atualmente, que tratasse de um banco de dados noSQL orientado a documentos, OpenSource e distribuído.

O documento irá falar de como ele foi desenvolvido, o motivo, suas funcionalidades, seu comportamento, algumas configurações, melhores práticas e quais são os produtos que se conectam nele para atender alguns cenários e necessidades existentes que precisam de velocidade de processamento em grandes massas de dados (Big Data).

O Elasticsearch

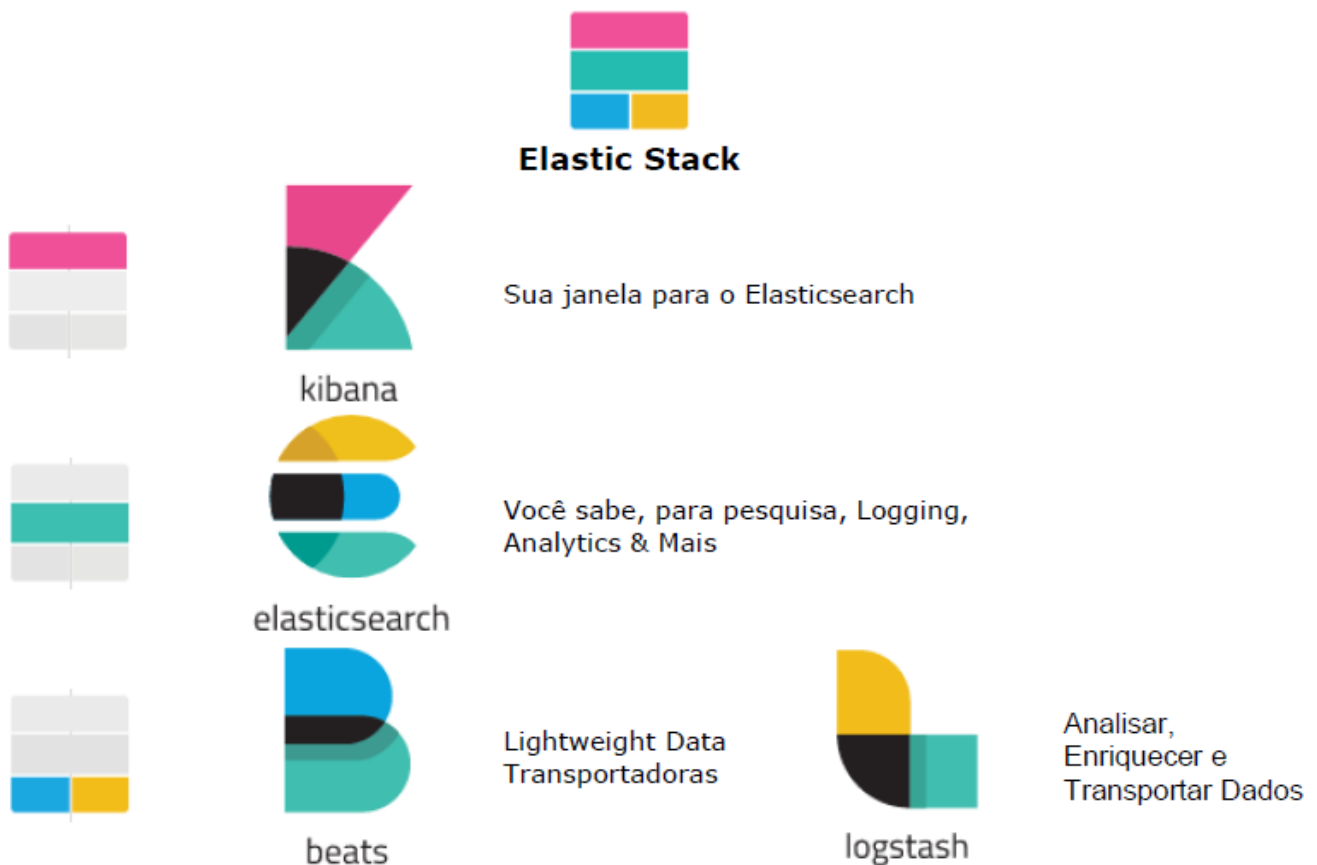
Elasticsearch, ou Elastic como é chamado atualmente, é um banco de dados noSQL orientado a documentos, OpenSource e distribuído. Ele foi criado a partir do Lucene*, um motor de busca em formato de biblioteca que permite implementar funções de busca em uma aplicação (linguagem nativa).

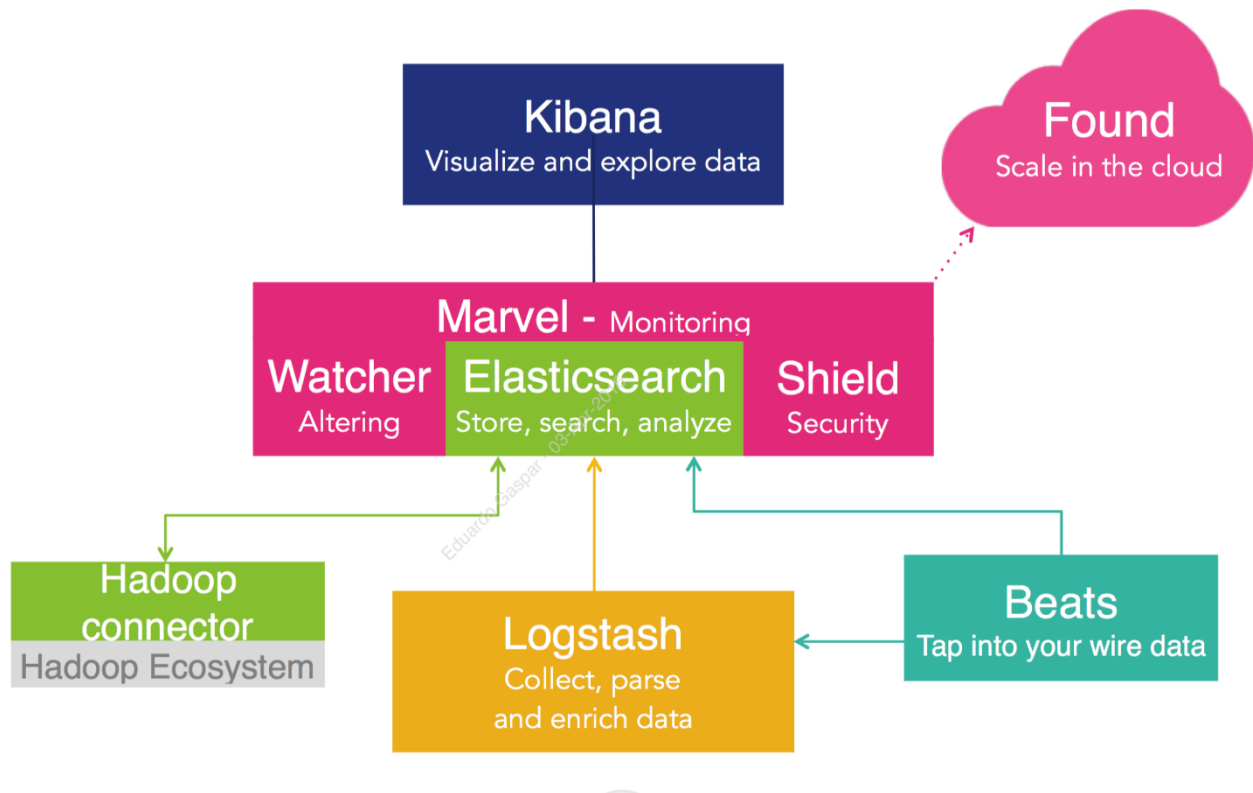
Lançado em 2010 porem com data oficial de fundação da companhia em 2012, baseado na linguagem Java se utilizando de JSON sobre HTTP, com o intuito de criar uma solução para pesquisa escalável.

Com ElasticSearch você é capaz de coletar dados ou transação, medir tendências, estatísticas, sumarização de anomalias utilizando parte do Logstash, Elasticsearch, Kibana e Beats. Podendo também executar pesquisas e extrair qualquer informação que seja de interesse. Outra funcionalidade é utiliza-lo como analytics ou business intelligence e investigar, analisar, visualizar executar consultas complexas de inteligência do negócio.

Segue abaixo uma demonstração ilustrativa da organização das soluções em torno do Elasticsearch, ou como o próprio fabricante diz “Elastic Stack” (Pilha Elastic).

Segue abaixo sua estrutura pura e logo após uma estrutura que utiliza o Elastic e demais componentes:





"Elasticsearch é um servidor de buscas distribuído baseado no Apache Lucene. Foi desenvolvido por Shay Banon[1] e disponibilizado sobre os termos Apache License. ElasticSearch foi desenvolvido em Java e possui código aberto liberado como sob os termos da Licença Apache."
Fonte: <https://pt.wikipedia.org/wiki/ElasticSearch>

Site oficial: <https://www.elastic.co/>

*Apache Lucene: <http://lucene.apache.org> e https://pt.wikipedia.org/wiki/Apache_Lucene

O Elasticsearch não suporta transações ACID assim como um banco relacional, isso tem implicações na arquitetura da aplicação e na arquitetura do ambiente de produção.

Outra diferença está na modelagem de dados, um banco de dados relacional suporta normalização pois trabalha muito bem com JOINS. Elasticsearch faz JOINS, mas devido a sua natureza distribuída, não foi muito bem desenvolvido para isso.

Existem diversas outras diferenças, principalmente porque Elasticsearch é uma tecnologia específica de RDBMS (Relational database management system) é uma categoria de banco de dados.

Mas as questões de transações ACID e modelagem são as principais diferenças.

As tabelas são conhecidas como Coleções e suas tuplas são conhecidos como Documentos, podendo cada documento ter quantos e qualquer atributo (colunas) independente entre as tuplas (linhas). Cada atributo (coluna) é definido por uma chave (Key) e seu valor (segundo as premissas de um documento JSON).

ACID

“ACID (acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade - do inglês: Atomicity, Consistency, Isolation, Durability), é um conceito utilizado em ciência da computação para caracterizar uma transação em um Banco de Dados, entre outras coisas.”

Fonte: <https://pt.wikipedia.org/wiki/ACID>

JSON

“JSON (com a pronúncia ['dʒejzən], J-son em inglês, (Jay-son)), um acrônimo para "JavaScript Object Notation", é um formato leve para intercâmbio de dados computacionais. JSON é um subconjunto da notação de objeto de JavaScript, mas seu uso não requer JavaScript exclusivamente.[1] [2] O formato JSON foi originalmente criado por Douglas Crockford e é descrito no RFC 4627. O media-type oficial do JSON é application/json e a extensão é .json.”

Fonte: <https://pt.wikipedia.org/wiki/JSON>

Ele pode ser visto como uma ferramenta exploratória para Big Data Analytics devido sua excelente performance. Possui desempenho no mesmo nível de outras soluções como Redshift, Spark, Presto, etc...

O link que segue abaixo possui um teste utilizando o Elasticsearch com informações sobre infraestrutura e grande volume dos dados para análise de uma informação.

Um bilhão de corridas de táxi em Elasticsearch

“Este post vai cobrir como eu levei um bilhão de + registros contendo seis anos de metadados de táxi em Nova York e analisou-los usando Elasticsearch em uma única máquina. [...]”

Fonte: <http://tech.marksblogg.com/billion-nyc-taxi-rides-elasticsearch.html>

O Elasticsearch trabalha através de uma API RESTful, com clientes em diversas linguagens (java, javascript, .net, php, etc).

Ele fornece uma Full Query DSL baseada em JSON. Que consiste em dois tipos de cláusulas:

- Leaf query clauses: procurar um determinado valor em um campo particular, tais como as match, term e range.
- Compound query clauses: são usados para combinar várias consultas de forma lógica (bool ou dis_max), ou para alterar o seu comportamento (constant_score).

Field Data Types

- Tipos de dados fundamentais
 - string, number, boolean, datetime, binary (base64)
- Tipos complexos
 - Array, objeto
- Outros tipos
 - Geo_point , geo_shape , ip , multi-field (útil no Kibana)

Nome	Valor Padrão	Descrição
_index		Onde o documento reside
_type		A classe de objeto que o documento representa
_id		O identificador exclusivo para o documento
_source	Default	Armazena o documento original que foi indexado
_all	Default	Índices de todos os valores de todos os campos do documento

Este exemplo retorna a conta numerada 20:

```
curl -XPOST 'localhost:9200/bank/_search?pretty' -d '{ "query": { "match": { "account_number": 20 } } }
```

Este exemplo retorna todas as contas que contenham o termo "mill" no endereço:

```
curl -XPOST 'localhost:9200/bank/_search?pretty' -d '{ "query": { "match": { "address": "mill" } } }
```

OBS: esse é exemplo é similar a query a seguir

```
SELECT state, COUNT(*) FROM bank GROUP BY state ORDER BY COUNT(*) DESC
```

E mais alguns exemplos genéricos para CRUD:

CREATE

```
curl -XPUT "http://server_name.com:9200/index/type/object_id" -d '{ <document data> }'
```

READ

```
curl -XGET "http://server_name.com:9200/index/type/object_id"
```

Para que resposta seja melhor visualizada usar o parametro "1?pretty" no final da linha, conforme exemplo:

```
curl -XGET "http://server_name.com:9200/index/type/object_id/1?pretty"
```



UPDATE

Necessário utilizar o termo “_update” após o Object ID:

```
curl -XPOST "http://server_name.com:9200/index/type/object_id/_update" -d '{<document data>}'
```

DELETE

```
curl -XDELETE "http://server_name.com:9200/index/type/object_id"
```

Bem-vindo ao wiki ElasticSearch-sql!

“Com este plugin você pode consultar ElasticSearch usando familiarizados sintaxe SQL. Você também pode usar funções ES em SQL.”

Fonte: <https://github.com/NLPchina/Elasticsearch-sql/wiki>

Requisitos e configurações

Memória

Uma máquina com 64GB de RAM é o ideal, porém máquinas de 32GB e 16GB também são comuns. Menos de 8GB não é recomendado e maior que 64GB terá problemas com a JVM perdendo performance.

CPU

A maior parte dos processos são de IOBound logo não demanda CPUs poderosas. É preferível o uso de CPUs com diversos núcleos para favorecer o processamento de concorrência entre os processos já que ele é Multithreading*.

Disco

Devido aos processos de IOBound é importante ter discos velozes. Unidades SSDs impulsionam o desempenho das operações de consulta e indexação. A empresa não recomenda o uso de NAS pois adiciona a latência de rede as operações, como o Elasticsearch faz replicação de dados nos outros nós do cluster. O uso de RAID 0 é uma forma de aumentar performance.

*Multithreading é a capacidade que o sistema operacional possui de executar vários threads simultaneamente sem que uma interfira na outra. Estes threads compartilham os recursos do processo, mas são capazes de ser executadas de forma independente.

Configuração geral

- JDK 1.8u20 + ou JDK 1.7u55 +
- ES_HEAP_SIZE
- Define o heap JVM (min e max mesmo valor)
- Regra é aprox. 1/2 da memória da máquina
- O restante da memória não está indo para o lixo!
- O sistema operacional irá usá-lo automaticamente para o cache do sistema de arquivos
- Prefere não alocar mais de 30gb
- Maior do que isso e a JVM não irá comprimir ponteiros de objeto mais (32 bits -> 64 bits)
- Nunca deixar a JVM entrar em swap
- Pode tentar bootstrap.mlockall ; melhor desligar de swap
- Em geral, mantenha os padrões de JVM

Nome do Cluster

- O padrão é Elasticsearch
- Mudar para algo significativo
cluster.name : < my_cluster_name >
- Todos os nós do cluster devem ter o mesmo nome do cluster

Nome do Nó

- O padrão é um nome de um super-herói da Marvel
- Mudar para algo significativo
node.name : < my_node_name >
- Certifique-se de todos os nós em um cluster tem um nome único
- Elasticsearch não se importa com o nome
bin / Elasticsearch --node.name = `hostname`

Portas

- HTTP
- A comunicação HTTP, por padrão, nas portas [9200-9300) (que vai automaticamente tentar e encontrar uma porta livre dentro da faixa)
http.port : 9200
- Transporte
- A comunicação entre nós, por padrão, nas portas [9300-9400) (que irá tentar encontrar um porto livre dentro da faixa automaticamente).
transport.tcp.port : 9300

Descoberta

- Usa o protocolo proprietário chamado zen
- Nós se encontram usando protocolo unicast por padrão
- Uma lista "originária" de nós para conectar-se é fornecida
discovery.zen.ping.unicast.hosts : [" host1 ", " host2 "]
- Multicast está disponível como um plug-in
- Plugin AWS e Azure disponível

Plugins

Existem três tipos de plugins:

- Web
 - Contem conteúdo web estático (Javascript , HTML e CSS)
 - Só precisa ser instalado em um nó
 - Não exigem reinicialização do nó para tornar-se visível
 - O conteúdo desses plugins é acessível através de uma URL como :
http:// yournode : 9200 / _plugin / [nome do plugin]
- Java
 - Contém apenas arquivos JAR
 - Deve ser instalado em cada nó no cluster
 - Exigem reinicialização para tornar-se visível
- Mistos
 - Contem os dois arquivos JAR e conteúdo web

Instalação

- A partir de repositório no site do Elastic
bin/plugin install marvel-agent
- A partir de URL direta ou caminho do sistema de arquivos
bin / plugin install file: ///< path_to_zip >/plugins/marvel-agent-2.1.1.zip
- Remover
bin/plugin remove marvel-agent

```
PUT marvel
{
  "settings" : {...},
  "mappings" : {
    "heroes" : {
      "properties" : {
        "name" : {
          "type" : "string",
          "analyzer" : "standard"
        },
        "color" : {
          "type" : "string",
          "index" : "not_analyzed"
        },
        "location" : {
          "type" : "geo_point"
        }
      }
    }
  }
}
```

Mapeamento para um tipo de documento específico

Obter os mapeamentos atuais de um tipo específico :

GET marvel/_mapping/heroes

Mapeamento Dinâmico

- Elasticsearch é "schema-less" por natureza
"Configurar somente quando você precisa"
- Não é necessário para definir mapeamentos para um índice
- Quando indexamos um novo documento, um mapeamento é criado dinamicamente para ele
- Extrair o máximo de informações a partir do documento JSON (por exemplo, tipos de dados) e aplicando padrões
- Por padrão , os mapeamentos têm uma natureza dinâmica
 - Campo "desconhecido" será adicionado ao mapeamento de tipo
 - Campo "desconhecido" é detectado de forma dinâmica
 - Boolean , duplos, por muito tempo, objeto , data, "analyzed" string
 - Dynamic true
- É possível restringir esse comportamento dinâmico
 - Dynamic false - campo "desconhecido" é ignorado (ainda existe em JSON)
 - Dynamic strict - campo "desconhecido" falhar pedido
- Definido no nível de tipo ou objeto

Bulk

- index / delete / update em vários documentos em um pedido
- minimizar chamadas
- número ótimo de documentos depende do caso e utilização específica

POST marvel/heroes/_bulk

```
{ "index" : { "_id" : "2" } }
{ "name" : "Thor", "strength" : 6 }
{ "delete" : { "_type" : "villains", "_id" : "1" } }
.
.
.
{ "create" : { "_index" : "dc", "_id" : "4" } }
{ "name" : "Batman", "intelligence" : 7 }
```

Cases

No site do Elasticsearch existe as seguintes referências de usuário: Uber, Netflix, PayPal, Github, Samsung, Huawei, etc.



Fonte: <https://www.elastic.co/use-cases>

No cenário nacional inclui a globo.com com os seguintes destaques listados abaixo.



Expectativa do usuário:

- Elimine o tempo de inatividade
- Garantia de todas as consultas gerar resultados

Aumentar as conversões em 8%

Suportar tráfego elevado:

- Servir 25 milhões de usuários únicos por dia
- Processar até 180 consultas por segundo

Entregar resultados em 100 ms



Usando Elasticsearch para garantir:

Unidade de e-commerce

Ativar vendedores para acessar dados críticos produto

Simplificar o gerenciamento de produto

Oferecer 24/7 alta disponibilidade

Garantir a escalabilidade

Sirva 4 milhões de vendedores

Lidar com o crescimento de 12 milhões para 20 milhões de listagens de produtos

Adicionar servidores em segundos, conforme necessário

Suas formas de utilizações mais populares:

Análise de dados de log

Os dados de eventos

Pesquisa de texto completo

Analytics e agregações

Visualização de dados com Kibana

Alertas e classificação

Motor de Sugestão

Pacote de dados / monitoramento de desempenho

Indexação

Os índices são particionados em shards para que possam ser distribuídos entre os múltiplos nós. O índice é um recipiente para dados sendo que o documento pode ser "indexado" e os resultados vão para um "índice". No Elasticsearch um shard é um único pedaço de índice e todos podem ser particionados (sharded), ou seja, sua importância se dá por permitir que você divida horizontalmente o volume de conteúdo. Ele permite que você distribua e paralelize as operações em toda a shard (potencialmente em múltiplos nós), aumentando assim o desempenho / rendimento.

Shard (arquitetura de banco de dados)

“Um shard de banco de dados, em português fragmento de banco de dados, é uma partição horizontal de dados em um banco de dados ou mecanismo de busca. Cada partição individual é referenciada como um shard ou shard de banco de dados. Cada shard é armazenado em uma instância de servidor de banco de dados separada, para distribuir a carga.

Alguns dados dentro de um banco de dados permanecem presentes em todos os shards, mas apenas alguns aparecem em um único shard. Cada shard (ou servidor) age como a fonte única para este subconjunto de dados.”

Fonte: [https://pt.wikipedia.org/wiki/Shard_\(arquitetura_de_banco_de_dados\)](https://pt.wikipedia.org/wiki/Shard_(arquitetura_de_banco_de_dados))

A quantidade de shards pode ser definida na criação da base de dados sendo 5 o valor padrão. Uma vez configurado este valor não pode ser modificado.

Os shards são divididos em primários, que aceitam a primeira escrita, ou réplicas, que repetem a escrita. Um índice deve ter ao menos 1 shard primário. As buscas podem ser feitas tanto nas shards primárias como nas réplicas.

Os shards são copiados por padrão para alta disponibilidade da informação. Em caso de falha uma réplica assumirá o posto de primária sendo que cada shard primária pode possuir de 0 a N réplicas associadas a ela. As réplicas dos shards são sempre alocadas em nós diferentes da primária e das outras réplicas.

As réplicas aumentam apenas o throughput de leitura.

Throughput

“Em redes de comunicação, como Ethernet ou packet radio, throughput, throughput de rede ou simplesmente taxa de transferência é a quantidade de dados transferidos de um lugar a outro, ou a quantidade de dados processados em um determinado espaço de tempo. Pode-se usar o termo throughput para referir-se a quantidade de dados transferidos em discos rígidos ou em uma rede, por exemplo; tendo como unidades básicas de medidas o Kbps, o Mbps e o Gbps.”

Fonte: <https://pt.wikipedia.org/wiki/Throughput>

Em um ambiente de rede / cloud em que as falhas podem ser esperadas a qualquer hora, é muito útil e altamente recomendável ter um mecanismo de failover no caso de um shard / node de alguma forma ficar offline ou desaparecer por qualquer motivo. Para este fim, Elasticsearch permite-lhe fazer uma ou mais cópias de fragmentos do seu índice.

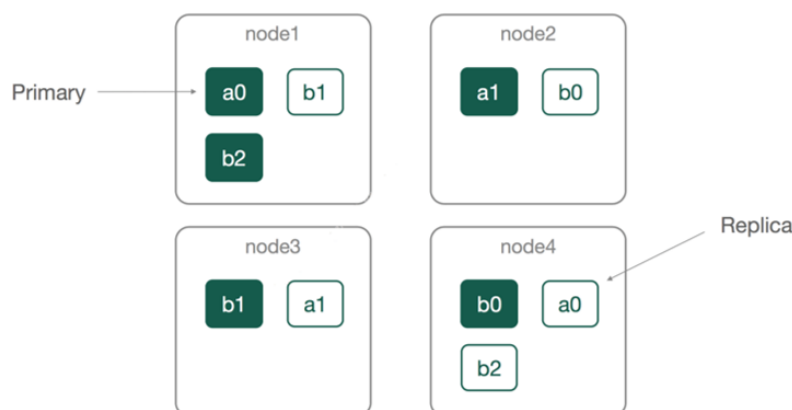
Com as diversas shards espalhadas pelo cluster o desempenho de consulta é alto além da otimização interna feita pela engine do Elasticsearch. Todavia, em caso de baixo desempenho é possível ligar o "query profiler" que permite analisar queries do ponto de vista da performance.

No entanto, os problemas de desempenho de uma query podem estar relacionados, muitas vezes, a problemas momentâneos de infra, neste caso o Elasticsearch provê uma API bastante extensa para monitoramento da saúde do cluster em diversos aspectos (CPU, memória, IO, etc.).

A disponibilidade de dados, caso servidores fiquem inacessíveis se dá através da replicação dos shards, e é feito tudo automaticamente pelo cluster (cópias automáticas, eleição automática de master node e primary shard, etc). Esse ainda é um foco de esforços da tecnologia.

A versão 2.x já melhorou muito sobre isso, todavia até que saia a próxima versão 5.x, a Elasticsearch ainda não recomenda utilizar o Elasticsearch como fonte primária de dados. Essa situação mudará no release 5.x (que deve acontecer por volta de julho/2016).

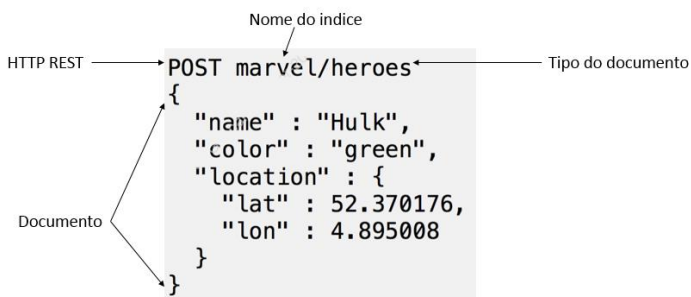
Exemplo de cluster com 4 nós:



- **Create/Index**
`PUT marvel/heroes/1`

```
{
  "name": "Hulk"
}
```
- **Read/Get**
`GET marvel/heroes/1`
- **Update**
`POST marvel/heroes/1/_update`

```
{
  "doc" : { "name": "The Incredible Hulk" }
}
```
- **Delete**
`DELETE marvel/heroes/1`



Index Templates

Muitas vezes vários índices de compartilham os mesmos mapeamentos e Templates de índice permitem criar Templates para vários índices correspondentes em seu nome com configurações e mapeamentos.

PUT _template/logstash ← Nome do template

```
{
  "template" : "logstash-*",
  "settings" : {
    "number_of_shards" : 1
  },
  "mappings" : {
    "_default_" : {
      "_all" : { "enabled" : false }
    }
  }
}
```

A expressão curinga, quando encontra o nome de um índice, irá aplicar automaticamente este modelo

- Apagando um template
curl -XDELETE localhost:9200/_template/te_prefix
- Buscando template
curl -XGET localhost:9200/_template/te_prefix
- Todos os templates
curl -XGET localhost:9200/_template

Multiple Index Templates

```
# create 2 templates
PUT _template/all
{
  "template" : "*",
  "settings" : {
    "number_of_shards" : 1,
    "number_of_replicas" : 0
  },
  "order":0
}

PUT _template/logstash
{
  "template" : "logstash-*",
  "settings" : {
    "number_of_shards": 2
  },
  "mappings" : {
    "_default_" : {
      "_all" : { "enabled" : false }
    }
  },
  "order" : 1
}
```

```
# index a document on new index
POST logstash-1/logstash/1
{
  "name":"elastic"
}

# verify final settings/mappings
GET logstash-1
```

```
{
  "logstash-1": {
    "mappings": {
      "_default_": {
        "_all": {
          "enabled": false
        }
      }
    },
    "settings": {
      "index": {
        "number_of_shards": "2",
        "number_of_replicas": "0",
      }
    }
  }
}
```

É permitido ter um apelido para um ou mais índices.

Sempre criar um alias

- Minimiza erros caso necessite apontar para múltiplos índices
- Permite que operações alterem o local de onde os pedidos e respostas vêm de sem modificar o código ou de configuração do aplicativo

Adicionando um alias:

```
POST _aliases
"actions" : [
  { "add" : { "index" : "employees", "alias" : "current" } }
]
```

Removendo um alias:

```
POST _aliases
"actions" : [
  { "remove" : { "index" : "employees", "alias" : "current" } }
]
```

Associando alias a múltiplos índices:

```
POST _aliases
{
  "actions" : [
    { "add" : { "index" : "logstash-2015-11-01", "alias" : "last_2_days" } },
    { "add" : { "index" : "logstash-2015-11-02", "alias" : "last_2_days" } }
  ]
}
```

Modificando alias atômicamente:

```
POST _aliases
{
  "actions" : [
    { "remove" : { "index" : "logstash-2015-11-01", "alias" : "last_2_days" } },
    { "add" : { "index" : "logstash-2015-11-03", "alias" : "last_2_days" } }
  ]
}
```

Os filtros podem ser associados com um alias

- Qualquer pesquisa executada contra o alias será automaticamente a ele

```
POST _aliases {
  "actions" : [
    {
      "add" : {
        "index" : "employees",
        "alias" : "sales",
        "filter" : { "term" : { "department" : "sales" } }
      }
    }
  ]
}
```


Busca

Permite encontrar documentos que correspondem a uma determinada consulta, além da consulta poder ser um parâmetro de URL ou um corpo de solicitação.

```
GET marvel/heroes/_search?q=hulk
```

```
GET marvel/heroes/_search
{
  "query": {
    "query_string": {
      "query": "name:hulk"
    }
  }
}
```

```
GET marvel/heroes/_search?q=name:hulk
```

- Clausula Match_all (select *)

```
GET /marvel/heroes/_search
{
  "Query" : {
    "Match_all" : { }
  }
}
```

- Implícita se não for fornecido

```
GET /marvel/heroes/_search
```

- Pontuação do documento será 1 na busca

Ela pode ser realizada em um ou mais índices.

/	search all types of all indices
/marvel	search all types of marvel
/marvel/heroes	search only heroes of marvel
/marvel,dc	search all types of marvel and heroes
/mar*	search all types of all indices that start with "mar"
/*/heroes	search heroes of all indices

Exemplo de Query e Filter

```

GET _search {
  "query": {
    "bool": {
      "must": [
        { "match": { "country": "United States" } },
        { "match": { "biography": "wealthy Genius" } }
      ],
      "filter": [
        { "range": { "birth_date": { "gte": "1985-01-01" } } },
        { "term": { "powers": "strength" } }
      ]
    }
  }
}
    
```

Cláusulas são usadas no query context, o que significa que eles são usados para **marcar o quão bem (score)** cada documento corresponde.

Ambas as cláusulas são usadas no filter context . Elas vão filtrar documentos que não correspondem , mas não afetam a pontuação para documentos correspondentes .

Agregação

- Permite agrupar os dados
 - Fornece agrupamento multi-dimensional dos resultados.
 - Ex.: Principais URLs por país.
- Muitos tipos disponíveis
 - Todos operam sobre os valores extraídos dos documentos
 - Geralmente a partir de campos específicos dos documentos, mas altamente personalizável através scripts
 - terms
 - range / date_range / ip_range
 - geo_distance / geohash_grid
 - histogram / date_histogram
 - Stats / avg / max / min / soma / percentis

Agregação

Filtrar por:

- ▼ Categoria
 - ☐ 1 estrela 17
 - ☐ 2 estrelas 82
 - ☐ 3 estrelas 72
 - ☐ 4 estrelas 57
 - ☐ 5 estrelas 9
 - ☐ Sem classificação 197
- ▼ Recepção 24 horas
 - ☐ Recepção disponível 24 horas 207
- ▼ Escolha seu tipo de propriedade
 - ☐ Hotéis 171
 - ☐ Apartamentos 166
 - ☐ Motéis americanos 41
 - ☐ Casas de temporada 21
 - ☐ Albergues 20
 - ☐ Pousadas campestres 9
 - ☐ Cama e Café (B&Bs) 6
- ▼ Avaliação dos hóspedes
 - ☐ Ótimo: 9 ou mais 7
 - ☐ Muito bom: 8 ou mais 99
 - ☐ Bom: 7 ou mais 179
 - ☐ Agradável: 6 ou mais 219
 - ☐ Sem classificação 18

The Westin St Francis San Francisco on Union Square ★★★★★

Muito bom 8,3
2.246 avaliações

Union Square, São Francisco – Perto do metrô

Sede do histórico Relógio Magneta, este hotel de luxo está localizado na Union Square (praça), bem em frente a um ponto de teleferico. Conta com várias opções gastronômicas no local.

Reservado 9 vezes hoje

Mostrar preços

Axiom Hotel ★★★★★

Fabuloso 8,7
169 avaliações

Union Square, São Francisco – Perto do metrô

O Hotel Axiom proporciona uma experiência com tecnologia avançada, incluindo WiFi gratuito de fibra ótica, fechaduras com sensores e check-in online. A propriedade possui um lounge-bar no local.

Reservado 11 vezes hoje

Mostrar preços

The Pickwick Hotel San Francisco ★★★

Bom 7,3
1.715 avaliações

Union Square, São Francisco – Perto do metrô

Situado a apenas 2 minutos a pé do Shopping Center Westfield, este hotel no centro de São Francisco oferece restaurante e bar.

Reservado 6 vezes hoje

Mostrar preços

Time Based Data

- Os eventos são representados como documentos
 - Normalmente, os eventos representam um evento do mundo real que ocorreu
- Cada evento (documento) está associado a um timestamp
 - Não pode ser repetido
 - Uma vez ocorrido, não pode ser modificado
- Documentos continuam a "fluir"
 - Escala com o tempo - sempre crescente
 - Impossível prever a escala de eventos futuros
- Dita a natureza de como usamos os dados baseados em tempo
 - Não podemos manter todos os eventos de todos os tempos
 - Os acontecimentos recentes são mais importantes que eventos antigos
 - Antigo é relativo ao problema que está sendo tratado

O Elasticsearch escreve para um índice por timeframe, suporta pesquisas cross-index sem prejuízo no desempenho e permite a "curadoria" de seus dados.

- Hora / dia / semana / mês / ano / timeframe customizado
- Configuração padrão: diário
 - Ex.: logstash - 2014/03/24
- Libera recursos dentro de seu cluster, ex.:
 - "Mover índices de mais de 30 dias para coldstorage"
 - "Fechar índices com mais de 60 dias"
 - "Excluir índices de mais de 90 dias"

Segurança

Atualmente a melhor forma, para não dizer única, de proteger o Elasticsearch é usando um outro produto chamado Shield, única parte que é cobrada pelo fabricante. Ele adiciona uma camada de autenticação e autorização de acesso que vai até as camadas mais baixas do sistema. O produto também adiciona outros recursos como auditoria, criptografia de acesso (porém, não criptografa dados em disco), controle de acesso por usuários e roles, integração com ldap, etc).

Proteja seus dados com Shield

Com a rápida adoção de Elasticsearch, é mais fácil do que nunca para armazenar, pesquisar e analisar seus dados. Escudo permite proteger facilmente esses dados com um nome de usuário e senha, além de simplificar sua arquitetura. Recursos avançados de segurança como criptografia, controle de acesso baseado em função, filtragem de IP, e auditoria também estão disponíveis quando você precisar deles. Seus dados são cada vez mais o seu bem mais valioso. A senha protege-o com Shield.

Fonte: <https://www.elastic.co/products/shield>

Os principais pontos da Shield no momento são:

- Autenticação: Proteja Elasticsearch com um usuário e senha

Protegendo os dados de modificação involuntária ou acesso não autorizado. Implementação de senha no cluster. Trabalha com Active Directory e suporte LDAP.

- Login e Gerenciamento de sessão no Kibana

Contém um plugin que fornece autenticação de usuário e suporte a sessão, tornando mais fácil para proteger o Kibana.

- Simplificar a sua arquitetura com proteção integrada

Evite construir, manter e testar uma solução de segurança externa. É totalmente integrado com Elasticsearch, verificando todos os pedidos e oferecendo desempenho sem sacrificar a segurança.

- Role-Based Access Control:

Permite a configuração de quem pode fazer o que no seu conjunto Elasticsearch. Criar uma conta de monitoramento para a equipe de TI / Operações, mas não pode acessar os dados. Conceder acesso só de leitura para seus usuários Kibana, ou mesmo suportar multitenancy, concedendo acesso apenas aos índices específicos.

- De campo e de nível de acesso ao documento

Obtém granularidade com o controle de acesso baseado em função em Elasticsearch. Permitindo restringir o acesso a campos individuais e impede que os usuários que acessam a documentos sensíveis com segurança em nível de documento igual a true.

- Comunicações & Criptografia com Filtragem de IP:

Encriptação SSL / TLS para proteger seus dados em utilização. Torna-se mais fácil para evitar espionagem ou adulteração por meio da criptografia tanto nó-a-nó a comunicações do cliente. Também permite evitar anfitriões não aprovados de se juntar ou de se comunicar com o cluster utilizando a filtragem de IP.

- Log de auditoria:

Ajuda a atender e superar uma variedade de regulamentos e exigências de segurança. Quer se trate de HIPAA, PCI DSS, FISMA, ISO, ou suas políticas internas, ele está coberto com um registro completo de toda a atividade do sistema e do usuário.

Logstash

O Logstash tem como objetivo a centralização de Processamento de Dados de todos os tipos, ele é um pipeline de dados que ajuda a processar registros e outros dados de eventos a partir de uma variedade de sistemas. Ele pode se conectar a uma variedade de fontes e dados de fluxo em grande escala a um sistema de análise central.

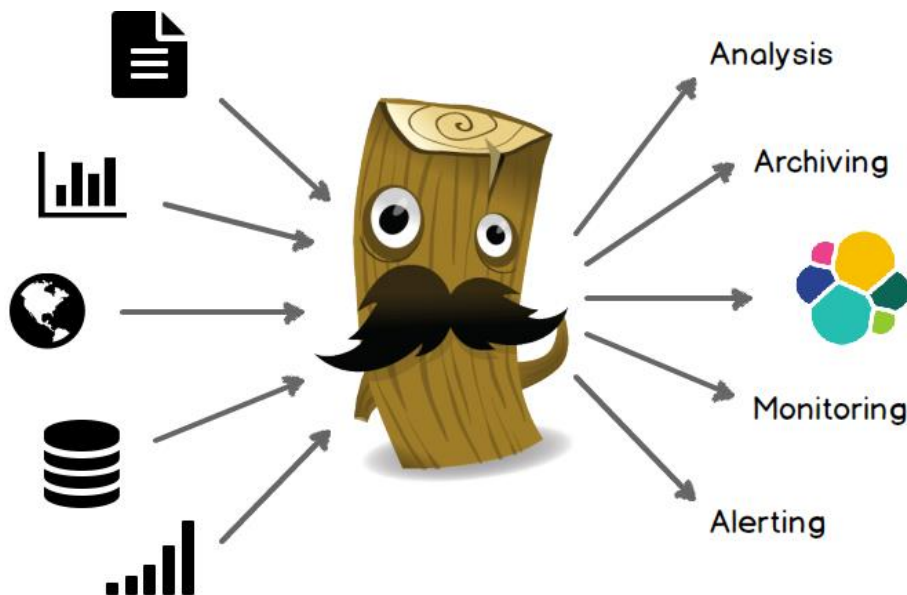
Ele normaliza variados Schema de dados críticos de negócios e que é muitas vezes espalhado entre sistemas diferentes, cada um no seu próprio formato. Ele permite analisar esses dados e os convergem para um formato comum antes de inseri-lo em sua análise de armazenamento de dados de escolha.

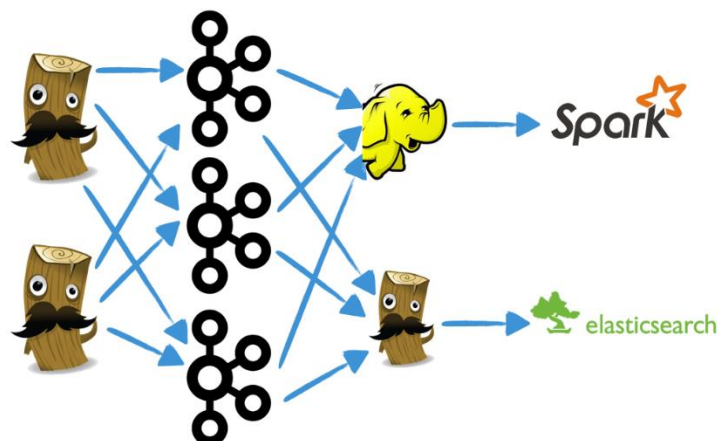
A maioria dos registros escritos por infra-estrutura e aplicações têm formatos personalizados. O Logstash fornece uma maneira conveniente e de lógica personalizada para analisar esses logs em grande escala.

Outputs tendem a ser categorizadas:

- Armazenamento: Elasticsearch, MongoDB, S3, File, etc.
- Relay: Redis, RabbitMQ, TCP, Syslog, etc.
- Notificação: PagerDuty, Nagios, Zabbix, etc.
- Metricas: Graphite, Ganglia, StatsD, etc.

Ele apresenta uma API para desenvolvimento de plugins.



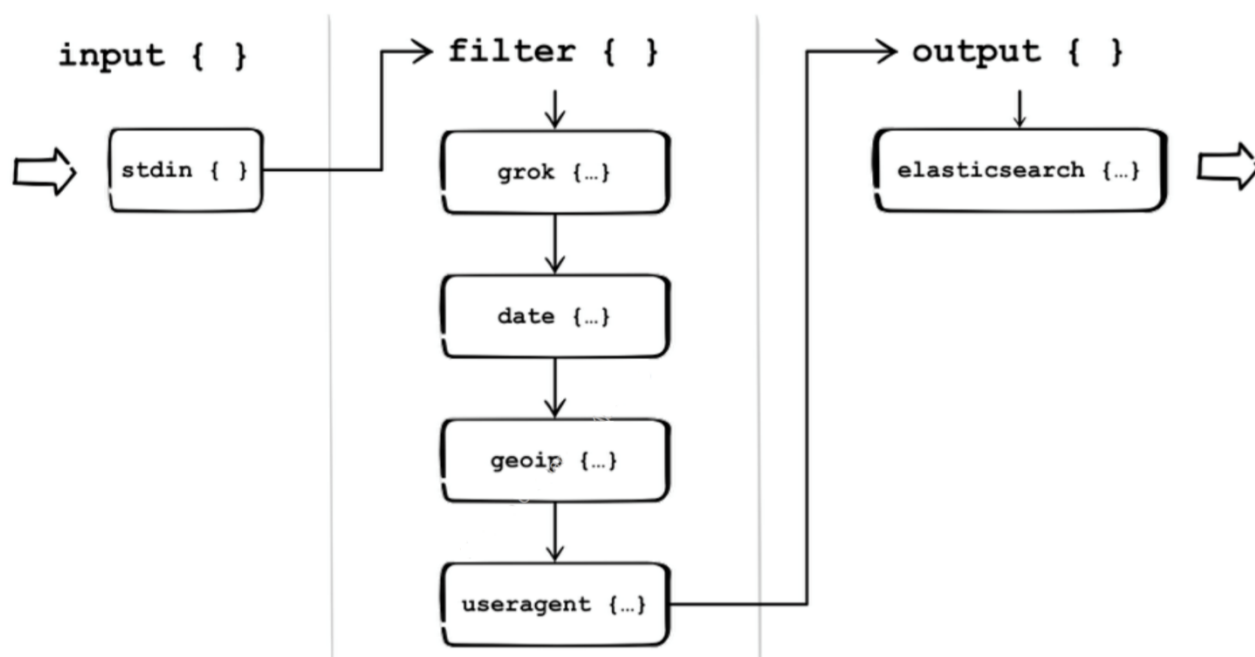


Exemplo de transformação do log:

```
66.249.73.185 - - [16/Feb/2014:09:47:54
-0500] "GET / HTTP/1.1" 200 37932 "-"
"Mozilla/5.0 (compatible; Googlebot/
2.1; +http://www.google.com/bot.html)"
```

```
timestamp => 2014-02-16T14:47:54.000Z
ip => 66.249.73.185,
geoip => {
  country   => "China",
  continent => "Asia",
  city_name => "Guangzhou",
}
...
```

Os filtros são processos em ordem conforme especificados.



```
{
  "message" => "83.149.9.216 - - [01/Jun/2015:21:13:45 -0200] \"GET /presentations/logstash-monitorama-2013/plugin/
notes/notes.js HTTP/1.1\" 200 2892 \"http://semicomplete.com/presentations/logstash-monitorama-2013/\" \"Mozilla/5.0
(Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36\""
}
```

grok {...}

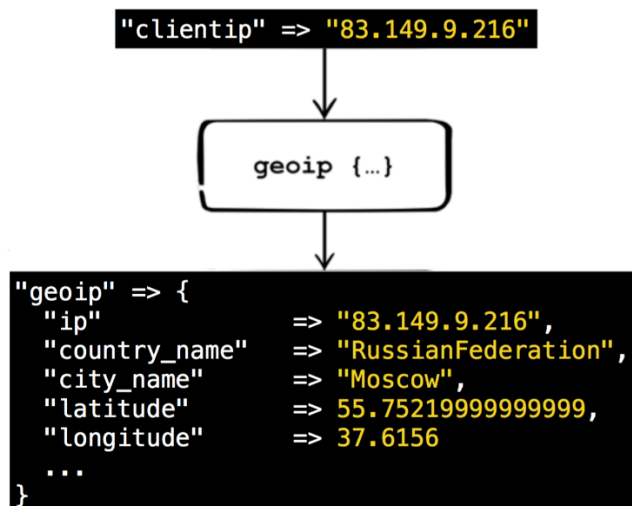
```
{
  "clientip" => "83.149.9.216",
  "ident" => "-",
  "auth" => "-",
  "timestamp" => "01/Jun/2015:21:13:45 -0200",
  "verb" => "GET",
  "request" => "/presentations/logstash-monitorama-2013/plugin/notes/notes.js",
  "httpversion" => "1.1",
  "response" => 200,
  "bytes" => 2892,
  "referrer" => "http://semicomplete.com/presentations/logstash-monitorama-2013/",
  "agent" => "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
32.0.1700.77 Safari/537.36"
}
```

Exemplos de configuração:

```
filter {
  grok => {
    match => {
      "message" => "%{IPORHOST:clientip} %{USER:ident} %{USER:auth}"
    }
  }
}
```

```
filter {
  grok => {
    match => {
      "message" => "%{IPORHOST:clientip} %{USER:ident} %{USER:auth} [%{HTTPDATE:timestamp}]
%{WORD:verb} %{DATA:request} HTTP/%{NUMBER:httpversion}
%{NUMBER:response:int} (?-| %{NUMBER:bytes:int}) %{QS:referrer} %{QS:agent}"
    }
  }
}
```

Controle de GeoPosição

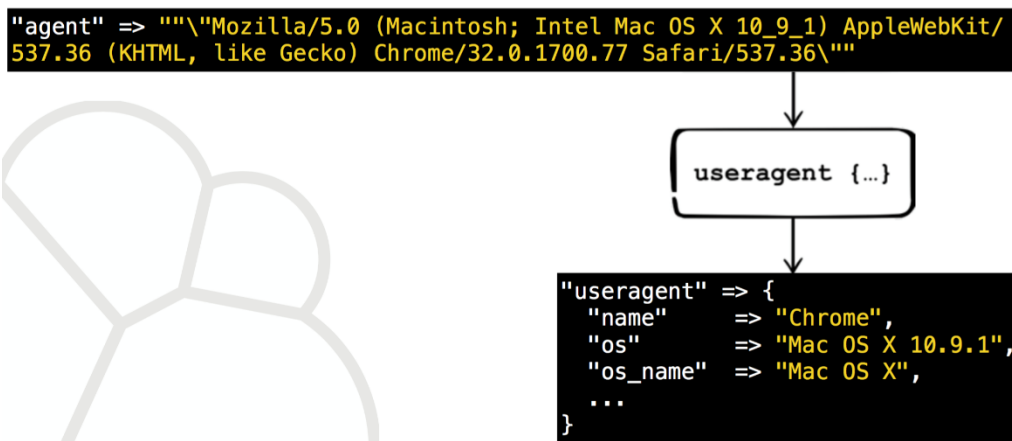


Configuração de GeoPosição

```

filter {
  geoip {
    database => ... # path to specific database
  }
}
  
```

Informação de Browser



Output para o Elasticsearch:


```
output {  
  elasticsearch {} # http://localhost:9200  
}
```

```
output {  
  elasticsearch {  
    document_id      => %{id}  
    document_type    => %{type}  
    index            => "apache_http_logs"  
  
    # logstash will round-robin over these hosts  
    hosts            => ["192.168.1.1", "192.168.1.2", "192.168.1.3"]  
    user             => "logstash"  
    password         => "password1"  
  
    # number of events buffered before sending a batch of events, default 500  
    flush_size       => 200  
  
    manage_template => false # will manage templates yourself, default true  
    workers         => 2  
  }  
}
```

Ele trabalha com condicionais para o processamento:

Operadores de comparação:

- igualdade: ==, !=, <, >, <=, >=
- regexp: =~, !~
- Inclusão: in, not in

Operadores booleanos:

- and, or, nand, xor

Operadores unários:

- ! (not)

Observações:

- Expressões podem conter expressões.
- Expressões podem ser negados com ! .
- Expressões podem ser agrupados com parênteses (...) .
- As expressões podem ser longas e complexas.
- Alerta para eventos Apache com 5xx resposta
- Alerta para eventos Apache com 4xx resposta
- Contar toda resposta de código apache via statsd
- Registrar todos os logs em ElasticSearch

Kibana

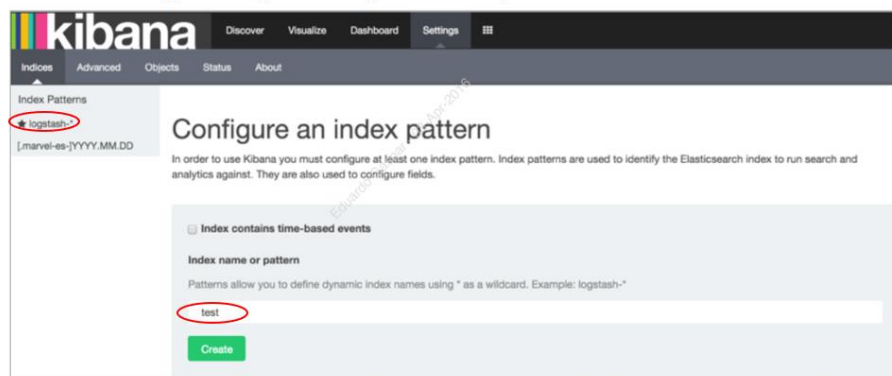
Arquitetado para trabalhar com ElasticSearch, Kibana dá forma a qualquer tipo de dados - estruturados e não estruturados - indexados em ElasticSearch. Tendo como benefício de poderosas capacidades de pesquisa e análise de ElasticSearch.

Ele permite criar gráficos de barras, gráficos de linha e dispersão, histogramas, gráficos de pizza, mapas, realizar transformações matemáticas, criar, salvar, compartilhar e incorporar dados para compartilhar visualizações e dashboards, sendo acessíveis com um navegador da Web (Chrome/Safari preferencial).



O usuário pode definir quais índices Kibana irá consultar, mudar como os campos são exibidos:

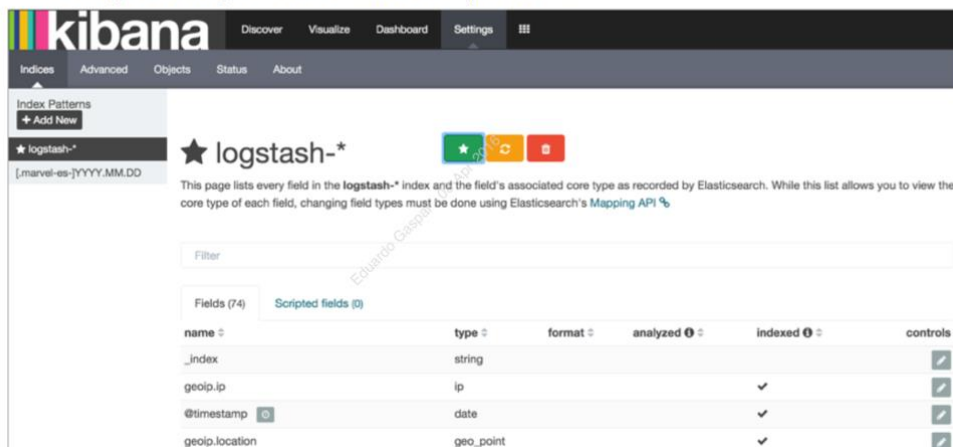
- Pode consultar um único índice (Ex.: "teste")
- Todos do logstash (Ex.: "logstash- *")



The screenshot shows the 'Configure an index pattern' page in Kibana. On the left, under 'Index Patterns', the 'logstash-*' pattern is selected. The main area contains the following elements:

- Index name or pattern:** A text input field containing 'test'.
- Create:** A green button to create the index pattern.

- Clique em índice para verificar os campos existentes
 - Na figura, logstash é o índice padrão



kibana Discover Visualize Dashboard Settings

Indices Advanced Objects Status About

Index Patterns
+ Add New

★ logstash-*

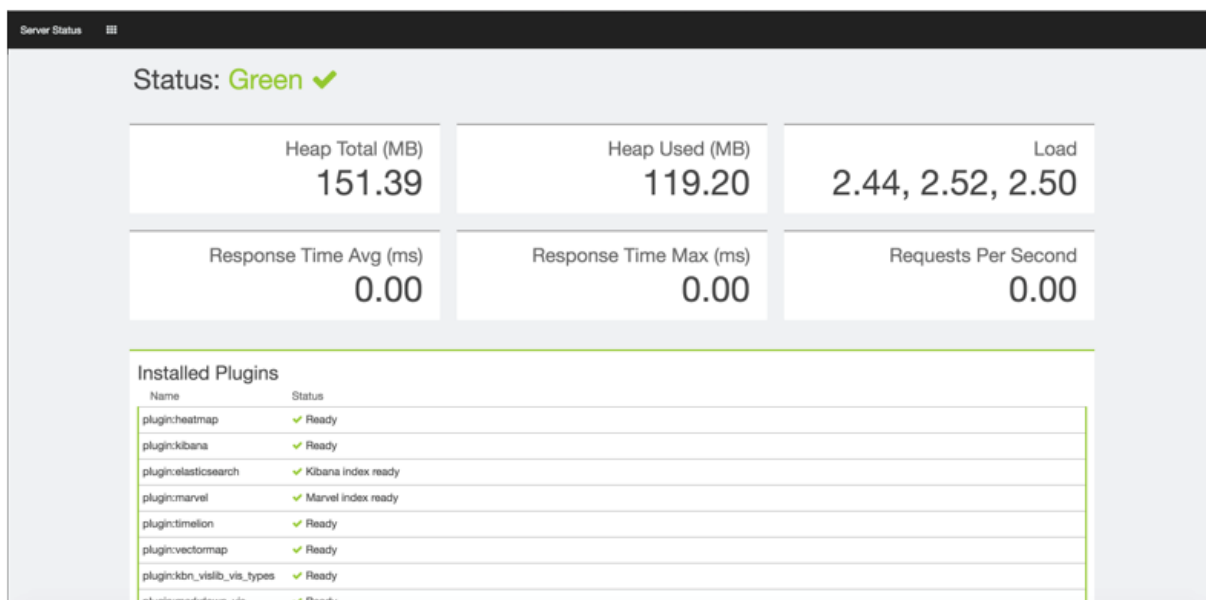
[marvel-es-]YYYY.MM.DD

This page lists every field in the **logstash-*** index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's [Mapping API](#).

Filter

Fields (74) Scripted fields (0)

name	type	format	analyzed	indexed	controls
_index	string				
geoip.ip	ip			✓	
@timestamp	date			✓	
geoip.location	geo_point			✓	



Buscas podem ser salvas

