Introdução à Elastic Stack

Gabriel Cestaro

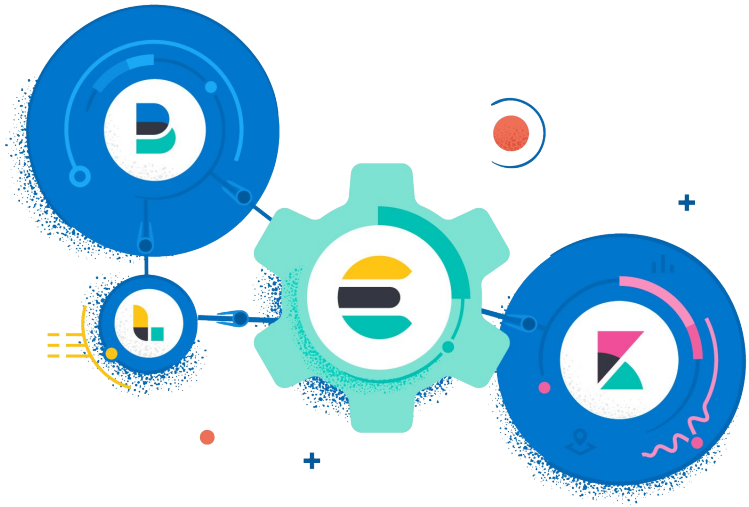
https://github.com/gcestaro/elasticsearch-hlrc-demo

Agenda

- O que é a Elastic Stack?
- Entendendo o Elasticsearch
- Mapeamentos e Indexação
- Buscas
- Como importar dados?
- Logstash
- Agregações
- Kibana
- Segurança e o X-Pack
- POC
- Referências

O Elasticsearch e a Elastic Stack

- Aplicação em Java rodando em cima do Apache Lucene
- RESTful API
- Elastic Stack
  - Elasticsearch, Logstash, Kibana, Beats, X-Pack, etc...
- Conceitos lógicos
  - Documentos (JSON) *
  - Índices
  - Índice invertido
    - Quebrando por palavras
    - *lowercase*
    - palavra no documento
      - posição + ID



* linhas

Em resumo

Data Collection

Data Aggregation & Processing

Indexing & storage

Analysis & visualization

Relevância de documentos

- R = TF * IDF
  - R: Relevance
  - TF : *Term Frequency* = Quão frequente um termo ocorre em um documento
  - IDF: *Inverse Document Frequency* = Quão frequente um termo aparece em todos os documentos

Relevância em um documento = Frequência no documento / Frequência em todos os documentos

Uso dos índices

- RESTful API: http(s)
- Client API: Java, Python, etc...
- Ferramentas de Analytics: Kibana, etc...

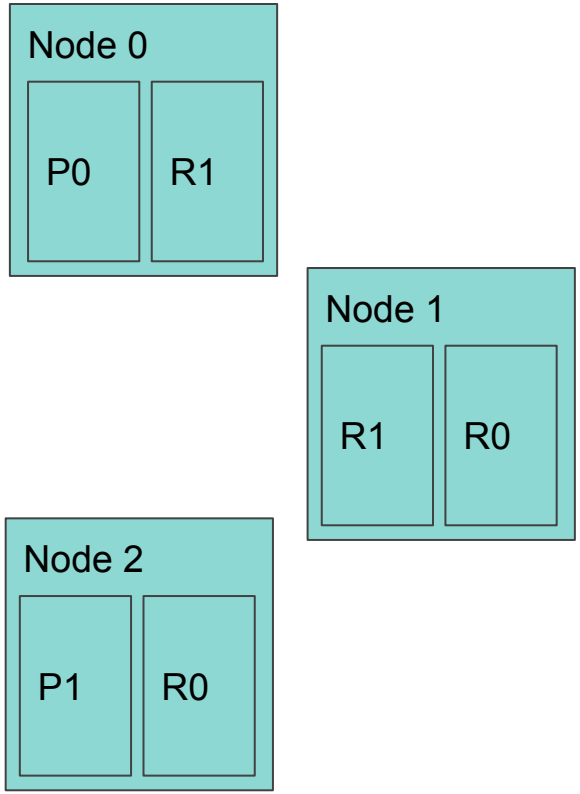
Novidades do ES7

- Tipos de documentos depreciados
- ES SQL
- Padrões para shards e réplicas
- Lucene 8
- Vários plugins para o X-Pack
- Java incorporado no pacote
- ILM - Index Lifecycle Management
- HLRC - High-level REST client para Java ()
- Melhorias de performance

Arquitetura ES

- Índices divididos em fragmentos (*shards*)
  - Instâncias do Apache Lucene
  - Podem estar em diferentes *Nodes* ou *Clusters*
- Shards primárias e réplicas.
  - Resiliência e tolerância a falhas
  - Sempre em número ímpar
  - Balanceamento de carga automático
  - A quantidade de shards não pode ser alterada sem reindexar
- Schema dos documentos é definido pelo mapeamento de campos no índice
  - Tipo *(string, byte, short, boolean, date)*
  - Índice do campo *(indexado para full-text search, analizado)*
  - Analyzer *(token e filtro de token)*

Shards - Exemplo com 2 primárias (P0 e P1) e 2 réplicas:

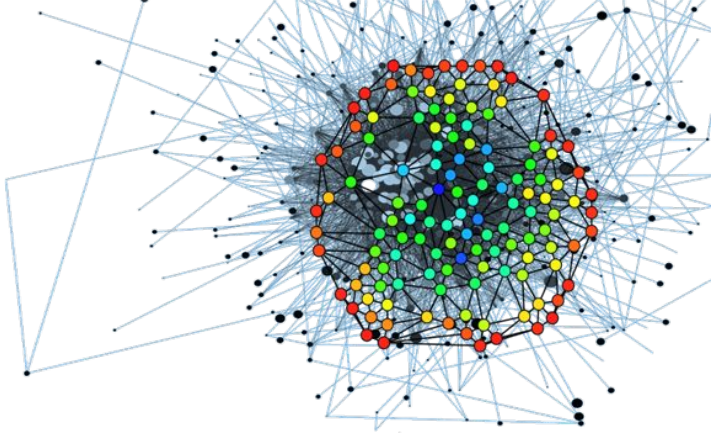


* O número de réplicas se aplica para cada primária! Portanto, P0 tem 2 réplicas e P1 também tem 2 réplicas, resultando em um total de 6 fragmentos (shards)

Analyzers

- Filtro de caracteres
  - Remoção de HTML
  - Conversão de símbolos (ex: “&” para “*and*” ou “e”)
- Tokenizer
  - Quebra de textos em espaços, pontuações, palavras, palavras sem letras
- Filtro de token
  - Alteração para letras minúsculas (*lowercasing*)
  - Normalizando variações da mesma palavra no plural, gerúndio, etc (*stemming*)
  - Sinônimos
  - Considerar ou não *stopwords* (ex: artigos e preposições)

Analyzers pré-definidos



- Padrão *(Standard)*
  - Removendo os limites, pontuações e deixando letras em minúsculo de palavras. (Boa escolha quando não se sabe o idioma do texto)
- Simples *(Simple)*
  - Quebra em qualquer caractere que não seja uma letra e deixa os caracteres em minúsculo
- Espaço em Branco *(Whitespace)*
  - Quebra em espaços em branco
- Idioma *(Language)*
  - Quebra em *stopwords* considerando a linguagem específica utilizada
  - *Stemming*

Atualizações de Documentos e Versionamento

- Documentos são imutáveis
  - Atualizações = cópia com novos valores + arquivos desatualizados marcados para deleção futura
- Alteração total
  - PUT ../indice
- Alteração parcial
  - POST ../indice/_update
- Delete por ID

Lidando com concorrência

- Controle com *lock* otimista
  - *_seq_no*
  - *_primary_term*
- *retry-on-conflict*

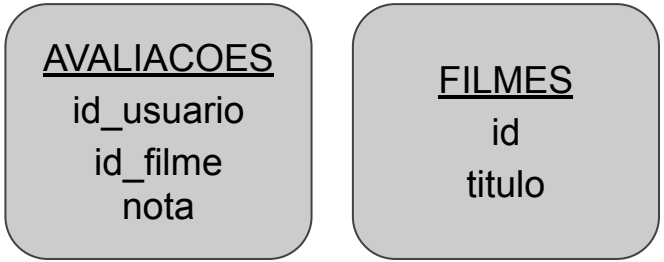
Full-text search

- Mapeamento de palavras-chave
  - Combinação exata
- Correspondência de tipo de texto (analyzer)
  - *Case insensitive*
  - *Stemmed*
  - *Stopwords* removidos
  - Sinônimos
  - Combinação de strings considerando os operadores lógicos OR ou AND

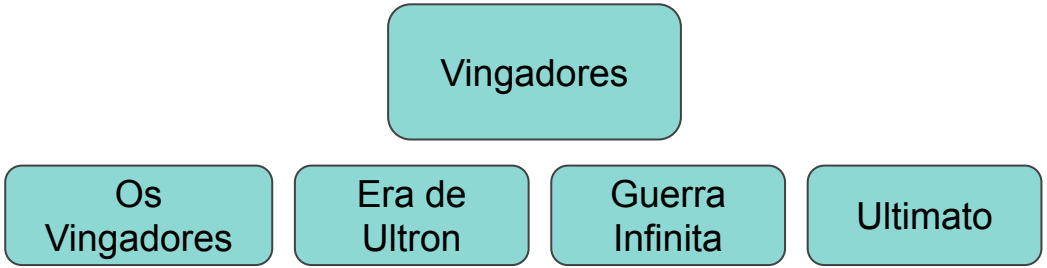
Modelando dados “relacionais”

Caso de Uso: Buscar avaliações de 0 a 5 para filmes.

Normalizado



Hierarquia de Filmes



Desnormalizado

* Mapeamento JOIN

Buscas - Query lite Search ou URI Search

${HOST}:${PORT}/{INDICE}/**_search**?{PARAMETROS_OPCIONAIS}

Exemplo: localhost:9200/filmes/_search?q=titulo:vingadores

Problemas:

- Criptografia
- Difícil debug
- Problemas de segurança
- Frágil

Onde usar?

- Experimentação, testes rápidos

Buscas - Request Body Search

```
GET .../indice/_search?pretty
{
        “query”: {
                “match”: {
                        “titulo”: “vingadores”
                }
        }
}
```

Buscas - Queries & Filters

- Filtros: expressão booleana
  - term: valor exato
  - terms: qualquer dos valores exatos
  - range: números ou datas em um intervalo
  - exists: verifica se um campo no documento existe
  - missing: verifica se um campo no documento não existe
  - bool: combinação de outros filtros com MUST, MUST_NOT e SHOULD
- Queries: retornam dados com base em relevância
  - match_all: busca em todos os documentos (padrão)
  - match: resultado analisado (exemplo: full-text search)
  - multi_match: idêntico a match, mas para vários campos
  - bool: combinação dos outros e considera relevância

** Filtros são mais rápidos e podem usar cache*
*** Filtros e queries podem ser combinados*

Buscas - Phrase Search

- match_phrase
  - Ex: “Vingadores - Era de Ultron”
    - “**match_phrase**” : {
      “titulo“ : “Vingadores Ultron”,
      “slop”: 3
      }

* slop é a distância em palavras entre uma palavra e outra, em qualquer direção

Buscas - Fuzzy Queries

Exemplo com o título: “Vingadores - Era de Ultron”

“Ving**o**d**a**res - Era de Ultron” -> distância de 2
“Vingador - Era de Ultron” -> distância de 2
“Vingadores - Era de Ultro**m**” -> distância de 1

- Uma forma de considerar erros de digitação
  - *The Levenshtein edit distance*
    - Substituição de caracteres
    - Inserção de caracteres
    - Remoção de caracteres
  - Auto *fuzziness*
    - 0 para palavras com 1 ou 2 caracteres
    - 1 para 3 à 5 caracteres
    - 2 para o restante

Buscas - Partial Matching

- Prefixo
  - {

```
"query" : {
        "prefix" : {
                "ano" : "201"
        }
}
```

  }

- Wildcard
  - {

```
"query" : {
        "wildcard" : {
                "ano" : "1*"
        }
}
```

  }

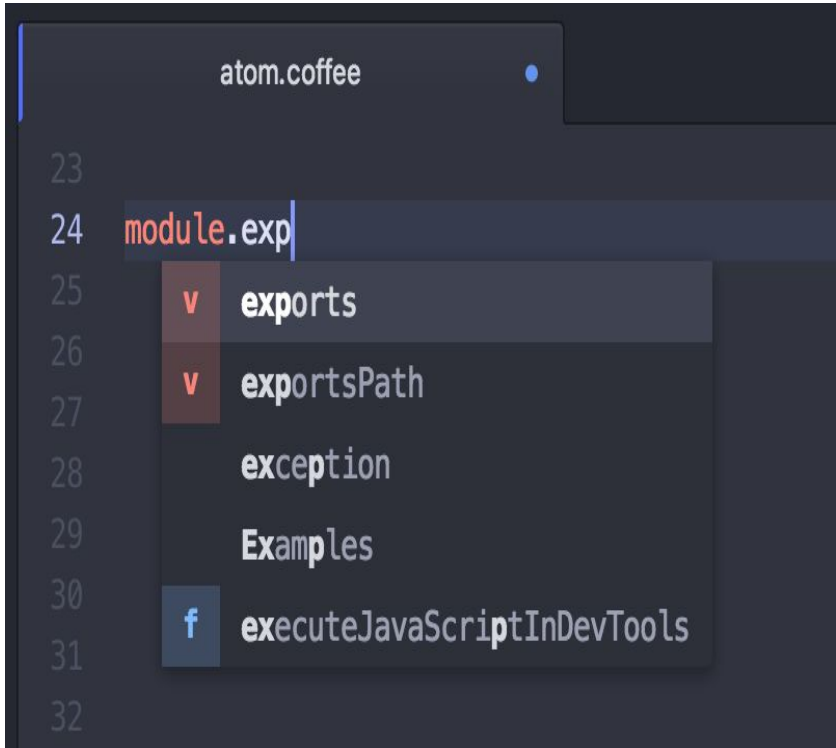
- Regexp
  - {

```
"query" : {
        "regexp" : {
                "ano" : "\d{4}"
        }
}
```

  }

Buscas - Search as you type (Query Time)

```
{
    "query" : {
        "match_phrase_prefix" : {
            "titulo" : {
                "query" : "Ultron",
                "slop" : 10
            }
        }
    }
}
```



* slop é a distância em palavras entre uma palavra e outra, em qualquer direção

Buscas - *Index-time & edge n-grams*

- “Ultron”
  - *Unigram*: [‘U’, ‘l’, ‘t’, ‘r’, ‘o’, ‘n’]
  - Bigram: [‘Ul’, ‘lt’, ‘tr’, ‘ro’, ‘on’]
  - Trigram: [‘Ult’, ‘ltr’, ‘tro’, ‘ron’]
  - 4-gram: [‘Ultr’, ‘ltro’, ‘tron’]
  - 5-gram: [‘Ultro’, ‘ltron’]
  - 6-gram: [‘Ultron’]

* Apenas a partir do começo da palavra
* Usado para sugerir complementos

Buscas - Criando analyzer para autocomplete

- PUT /filmes

```
{
    "settings": {
        "analysis": {
            "filter": {
                "filtro_autocomplete": {
                    "type": "edge_ngram",
                    "min_gram": 1,
                    "max_gram": 20
                }
            },
            "analyzer": {
                "autocomplete": {
                    "type": "custom",
                    "tokenizer": "standard",
                    "filter": [ "lowercase", "filtro_autocomplete" ]
                }
            }
        }
    }
}
```

Paginação

- From (começa com zero)
  - a partir de qual registro
- Size
  - quantos registros

* Paginação profunda pode prejudicar o desempenho
* Todo resultado deve ser recuperado, coletado e ordenado
* Considerar um limite para quantos registros retornar ao usuário

Ordenação

- ../indice/_search?sort=avaliacao&***titulo***
- Ordenação não funciona em strings analisadas por causa do índice invertido
  - Solução: criar uma palavra-chave com o valor original do campo

```
PUT .../indice
{
        "mappings": { "properties" : { "titulo": { "type" : “text", "fields": {
                                                                "raw": {
                                                                        "type": “keyword“
                                                                }
        } } } }
}
```

Importando vários documentos (Bulk)

Divisão em duas linhas para ajudar o ES a balancear entre as várias shards e nodes

1. Operação + Índice + _id
2. Documento

{ "create" : { "_index" : "series", "_id" : "1", "routing" : 1} }
{ "id": "1", "film_to_franchise": {"name": "franchise"}, "title" : "Star Wars" }

{ "create" : { "_index" : "series", "_id" : "260", "routing" : 1} }
{ "id": "260", "film_to_franchise": {"name": "film", "parent": "1"}, "title" : "Star Wars: Episode IV - A New Hope", "year":"1977" , "genre":["Action", "Adventure", "Sci-Fi"] }

Como importar dados?

- Scripts standalone bulk via REST API
- Logstash
- Beats
- Lambda AWS
- Kinesis Firehose
- Kafka
- Spark

Logstash

- Análise, transformação e filtros de dados
- Criar estruturas de dados a partir de dados não estruturados
- Anonimizar dados sensíveis ou ignorá-los
- Localização geográfica
- Escala em vários nós
- Garante ao menos uma entrega
- Absorve o *throughput* de picos de carga

Agregações

- Métricas
  - Média
  - Status
  - Min/max
  - Percentual
- Buckets
  - Histogramas
  - Intervalos
  - Distâncias
  - Termos significativos
- Pipelines
  - Médias móveis
  - Média de buckets
  - Soma acumulada
- Matriz
  - Status em matrizes (2d)

Kibana

- UI para visualizar agregações
- Dev tools
- Configurações
- Dashboards

Segurança e o X-Pack

- Controle de acessos
  - Baseado em funções + privilégios por função + usuários e grupos com funções
  - Índices, *aliases*, documentos e/ou campos
  - Possível usar Active Directory ou LDAP com DNS para usuários
- Integridade de dados
- Auditoria

Referências

- https://lucene.apache.org/
- https://www.elastic.co/guide/index.html
- https://www.elastic.co/pt/downloads/elasticsearch
- https://www.elastic.co/pt/downloads/logstash
- https://www.elastic.co/pt/downloads/kibana
- https://www.udemy.com/course/elasticsearch-7-and-elastic-stack/
- https://www.alura.com.br/curso-online-elasticsearch
- https://www.alura.com.br/curso-online-elasticsearch-introducao
- https://www.alura.com.br/curso-online-elasticsearch-analise-consulta-dashboard

Imagens

- https://s.wincalendar.net/img/pt/dias/18-junho-2020.png
- https://static-www.elastic.co/v3/assets/bltefdd0b53724fa2ce/blt9d2163a1b0c666f2/5d26506ae802da1244b26bf4/illustration-stack-data-flow-header.png
- https://www.oxigenweb.com.br/wp-content/uploads/2016/04/TARGET1.jpg
- https://logz.io/wp-content/uploads/2018/08/image21-1024x328.png
- https://ecuinc.biz/wp-content/uploads/2017/11/index.jpg
- https://joebalestrino.com/wp-content/uploads/2019/02/Marketplace-Lending-News.jpg
- https://www.gre.ac.uk/__data/assets/image/0011/1191953/analysis-banner.jpg
- https://casis.llnl.gov/content/assets/images/graph.png
- https://cdn4.wpbeginner.com/wp-content/uploads/2015/05/updated.jpg
- https://blog.theodo.com/static/309ba67d50711a0f56e05f871afaf358/83a78/tartan-track.jpg
- https://our.umbraco.com/media/wiki/25126/635092270378752912_Full-Text-Search_128.PNG?height=154&width=281&bgcolor=fff&format=png
- https://dev.observatoriodocinema.bol.uol.com.br/wp-content/uploads/2019/05/thorgordo-1.jpg
- https://flight-manual.atom.io/using-atom/images/autocomplete.png
- https://image.freepik.com/fotos-gratis/informacoes-paginas-livro_19-128050.jpg
- https://2.bp.blogspot.com/_cM2I6Y3UIgg/SQ-4Cv5qy6I/AAAAAAAAA9Q/pq-XIOZ983E/s320/dados.jpg
- https://assets.zabbix.com/img/brands/logstash.svg
- https://www.kindpng.com/picc/m/544-5447437_kibana-logo-png-transparent-png.png
- https://lh3.googleusercontent.com/proxy/luHNj6TZSbKHcsLPf8-ssyXvdASrjNHveco74FQqMTMw_ADf6ZZsI0S4ClKIrzRjjS_i7QjIj65x0c4rt51wo8vkI-pg9PgL1MGaEPytXKKML5oHBDIrm_a_v6La9w