

A aula interativa do Módulo 4 – Bootcamp Engenheiro de Dados em breve!

Importante:

- 1) Não se esqueça de acessar a aula com seu e-mail cadastrado no ambiente de aprendizagem do IGTI e seu nome completo. Entrou com os dados errados? Saia da sala e entre novamente com os dados corretos.
- 2) Para sua frequência ser computada, responda a enquete no ambiente de aprendizagem, no horário indicado pelo professor. A enquete ficará disponível por 10 minutos.
- 3) Utilize o Chat para interagir com os colegas durante a aula, a ferramenta Raise Hands para pedir a palavra, e, em caso de dúvidas sobre o conteúdo, utilize o Q&A (perguntas e respostas) para que o professor tutor possa respondê-las.



Infraestrutura de Dados e Arquitetura Escalável

PRIMEIRA AULA INTERATIVA

PROF. GUSTAVO AGUILAR



Gustavo Aguilar



Formação



- Bacharelado em Ciência da Computação (PUC-Minas).
- Pós-graduação em Administração de Banco de Dados (Newton).
- Especialização em Docência do Ensino Superior (Newton).
- MBA em Ciência de Dados (IGTI).
- Instrutor Certificado Microsoft (MCT).



Atuação



- Administração de Bancos de Dados.
- Ambientes de missão crítica em diversas plataformas.
- Modelagem, Arquitetura e Engenharia de Dados.
- Persistência e Pesquisa de Dados.
- Sistemas Distribuídos e Cloud Computing .
- Metodologias Ágeis e DevOps.
- Coordenador e professor de MBA e Bootcamp.



Nesta aula



- ☐ Pesquisa Indexada.
- ☐ Elasticsearch.
- ☐ Dúvidas do Trabalho Prático.



Pesquisa Indexada



- **Projetos de Big Data / BI:** manipulação e pesquisa em volumes gigantescos de dados, mudança dinâmica de regras e análises realtime.
- **E-commerce:** sugestões de produtos relacionados, recursos de autocompletar, personalização de produtos e promoções em tempo real de acordo com o perfil do cliente etc.
- **Buscadores de preços:** informar para um usuário quando algum produto atingir o preço configurado por ele.
- **Tendências de mercado:** prever comportamentos e desejos dos usuários, antecipar procuras ou compras.
- **Grande lacuna** de ferramentas de pesquisa indexada:
 - Mais robustas, escaláveis e dinâmicas → “pesquisa elástica”.

Ferramentas de Mercado



- Apache → **Solr**:



- Abstração construída em cima do Lucene.
- API HTTP trocando dados com XML/JSON, permitindo usar a busca e consumir o serviço pronto pela web.
- Filtragem e pesquisa gerenciada (auxiliadores de busca como sugestão).
- Pesquisa geoespacial, cache com atualização incremental.
- Distribuição e replicação de dados.

Ferramentas de Mercado



- Elastic → **Elasticsearch**:
 - Concorrente direto do Solr, com os mesmos recursos e além.
 - Mais moderno, com mais facilidades e atualização dos índices próxima do tempo real.
 - Alta disponibilidade e capacidade para tratar grandes volumes de dados.
 - Pesquisas full-text, geográficas e analíticas.
 - Disponibiliza uma API REST full.
 - Possibilidade de implementação de clusters, sendo totalmente escalável.



Ferramentas de Mercado



- **Amazon Elasticsearch Service:**
 - Serviço gerenciado que facilita a implantação, a operação e o dimensionamento de clusters do Elasticsearch na nuvem (AWS).
 - Recursos para dimensionar, controlar segurança, estabilidade e integração com outros serviços:
 - Amazon CloudWatch (para monitoramento).
 - AWS CloudTrail (para auditoria).
 - Amazon S3 (armazenamento).
 - DynamoDB (banco de dados).
 - Escalabilidade nativa de computação em nuvem.



ElasticSearch Service



Ferramentas de Mercado

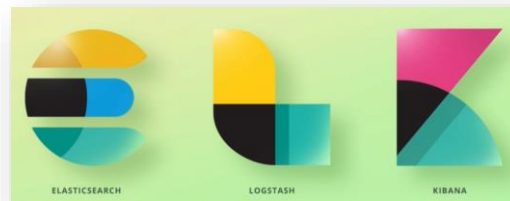
- Microsoft → **Azure Search**:



- Serviço oferecido na nuvem (Microsoft Azure).
- Funcionalidade exposta por meio de uma API REST ou SDK do .NET.
- Integração nativa com o Azure Blob Storage (armazenamento).
- Pesquisa cognitiva, geográfica e análise linguística.
- Modelagem de relevância (pontuação) dos valores.
- Escalabilidade nativa de computação em nuvem.

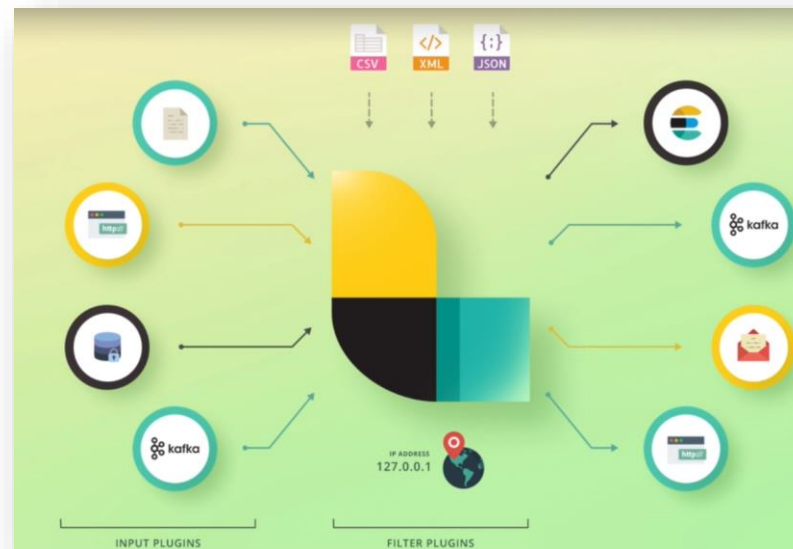
Elasticsearch

- Parte integrante do **Elastic Stack**:
 - Grupo de produtos de código aberto da Elastic.
 - Coletar dados de qualquer tipo de fonte e em qualquer formato.
 - Pesquisar, analisar e visualizar esses dados em tempo real ou não.
 - Inicialmente conhecido como **ELK Stack**.
 - Letras representavam as iniciais do nome dos primeiros produtos do pacote:
Elasticsearch, Logstash e Kibana.



Elastic Stack

- **Elasticsearch (E):** mecanismo de pesquisas e análises distribuído.
- **Logstash (L):** ferramenta de processamento de dados.
 - Coletar dados de várias fontes.
 - Transformar dados.
 - Enviar dados.
 - Filtros predefinidos.
 - Mais de 200 plugins.



Elastic Stack

- **Kibana (K):**
 - Ferramenta de visualização e exploração de dados.
 - Oferece gráficos interativos e fáceis de usar.
 - Agregações e filtros pré-construídos.
 - Suporte geoespacial.
 - Alta integração com o Elasticsearch.

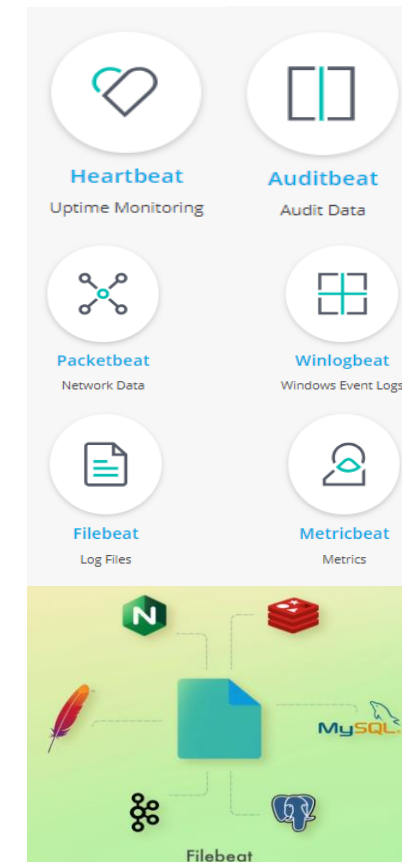
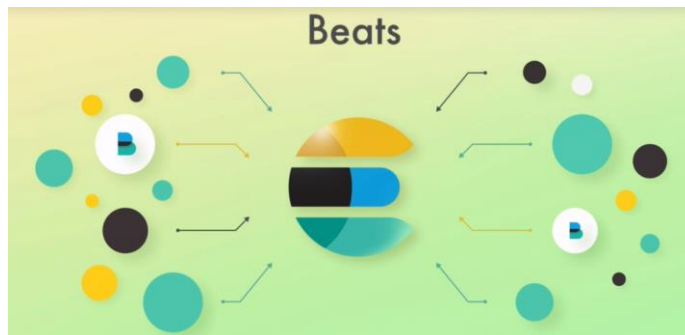


Elastic Stack



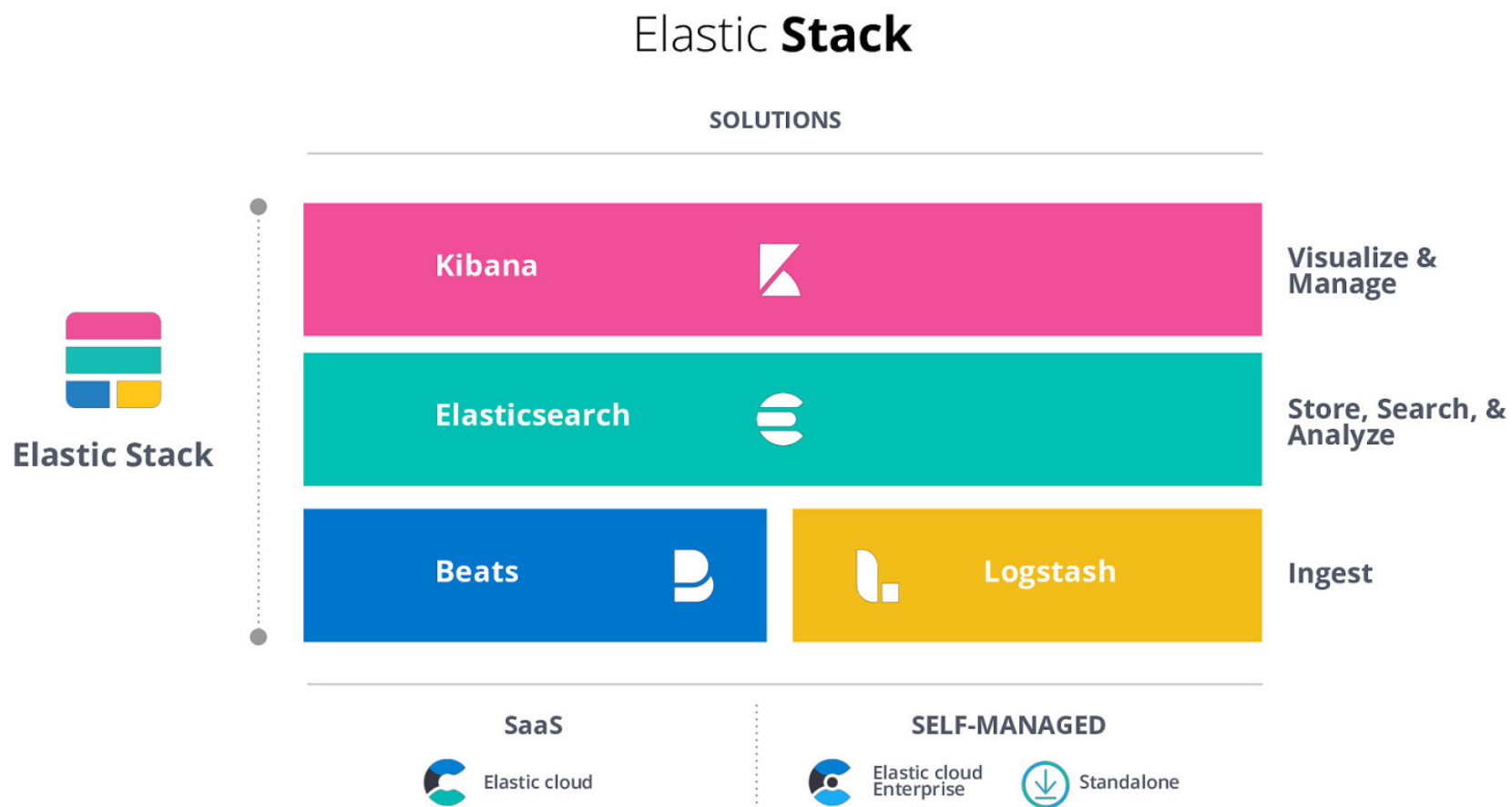
- **Beats:**

- Quarto produto adicionado posteriormente ao pacote.
- Plataforma com o propósito único de coletar dados (push / pull).
- Envio de dados diretamente para o Elasticsearch ou através do Logstash (quando precisar tratar o dado antes).
- Diversos pacotes de métricas (SO, BD, Network etc.).



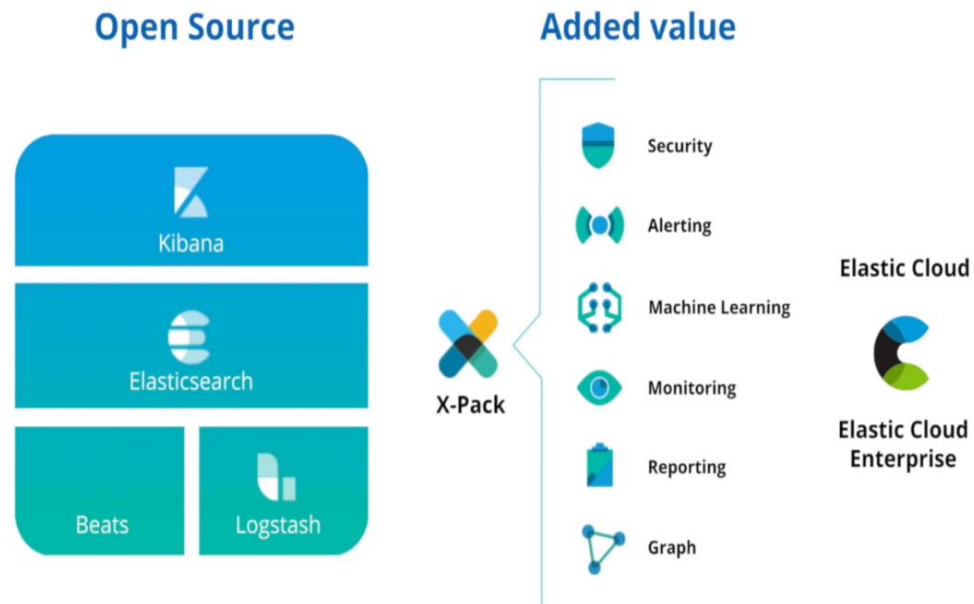
Elastic Stack

IGTi



Elastic Stack

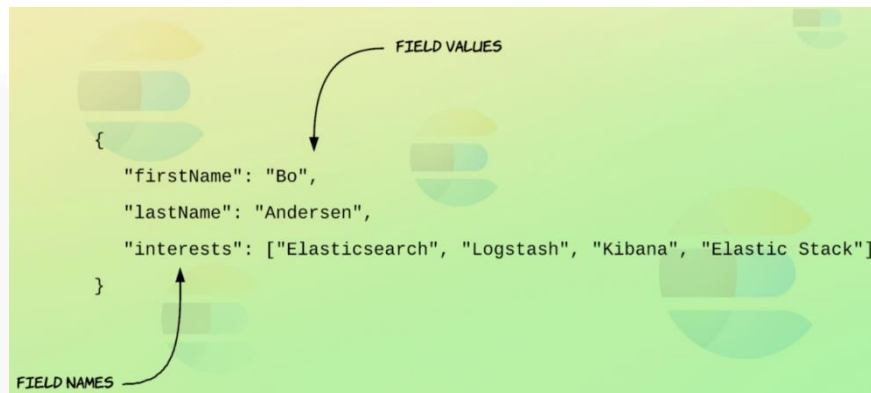
- **X-Pack:** conjunto de produtos não gratuitos, desenvolvidas pela Elastic, que prometem agregar valor ao pipeline de dados:



Elasticsearch



- **Dados são armazenados como documentos** no formato **JSON**.
- Documento corresponde ao conceito de linha (tupla) em SGBDR.
- Documento contém **campos (*fields*)** e **valores**:
 - Conceito correspondente ao de tabela em um banco de dados relacional.



Elasticsearch



- Cada documento possui seu ID:
 - Identificador único do documento dentro de um índice.
 - Pode ser atribuído automaticamente ou manualmente via código.
- Um documento é uma **unidade básica de informação** que pode ser **indexada**.
- Para armazenar os documentos ➔ **índice** (*index*).
- Os índices são o cerne da engine de armazenamento e busca do Elasticsearch.

Elasticsearch

- **Índices:**

- Coleção de documentos com características similares.
- Identificados por um nome.
- Nome em letras minúsculas.
- Nome usado nas operações.
- Correspondente ao conceito de database.
- Ex.: índice para dados do cliente.



- Dentro de um índice: armazenar quantos documentos desejar ou couberem no local de armazenamento de dados do Elasticsearch.

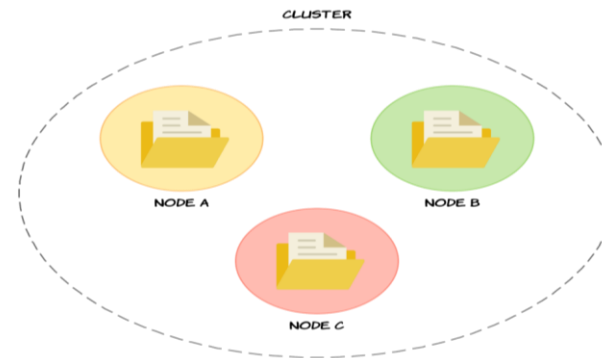
Elasticsearch - Arquitetura

- **Nó (*node*):**

- Servidor (físico ou virtual) que armazena dados.
- Instância do Elasticsearch.

- **Cluster:**

- Coleção de nós.
- Armazena e contém o conjunto de dados inteiro do cluster Elasticsearch.
- Cada nó participa das operações de indexação e pesquisa do cluster quando uma determinada operação envolver os dados que ele armazena ou armazenará.



Elasticsearch - Arquitetura

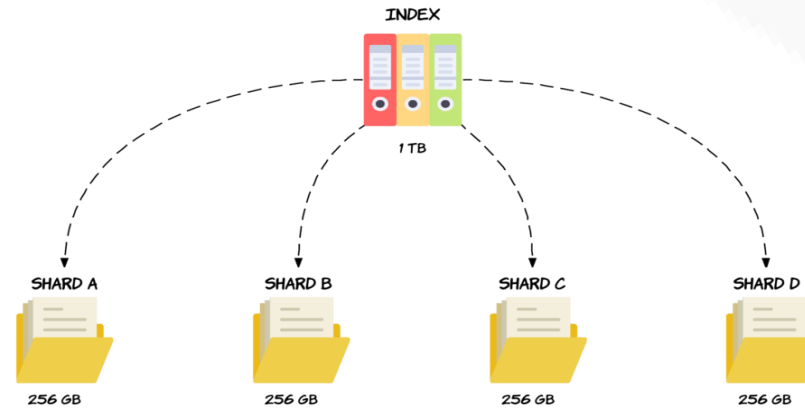
- Cluster expõe uma **REST API HTTP**:
 - Qualquer nó do cluster pode manipular solicitações HTTP.
- Cada nó dentro do cluster possui informações dos demais nós:
 - Capaz de encaminhar solicitações usando a camada interna de transporte.
- Todo cluster possui um **nó mestre (*master*)**:
 - Qualquer nó pode ser designado para ser o nó mestre por padrão.
 - Nó responsável por coordenar as alterações no cluster, como adicionar ou remover nós, criar ou remover índices etc.
 - Responsável por atualizar o estado do cluster.

Elasticsearch - Arquitetura



- **Sharding:**

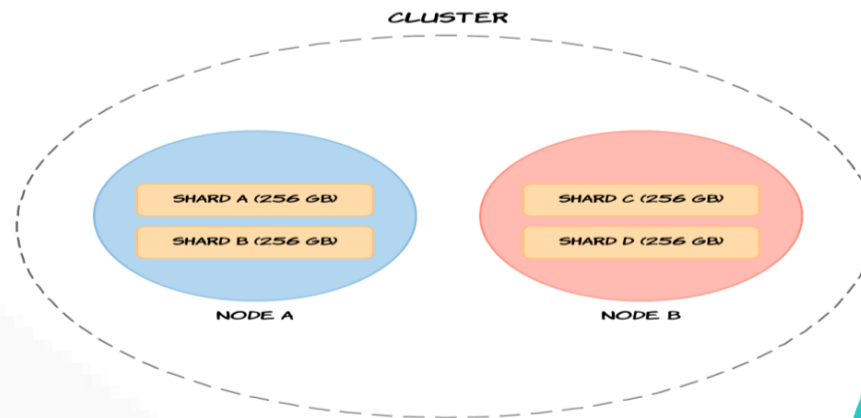
- Para **escalar** na topologia em cluster.
- Particionamento horizontal de dados.
- Divide os índices em partes menores:
 - **Shards (fragmentos).**
- Um shard só contém um subconjunto de dados de um índice.
- Um documento é armazenado em apenas um dos shards:
 - Não há divisões (*splits*) no nível de documentos.



Elasticsearch - Arquitetura

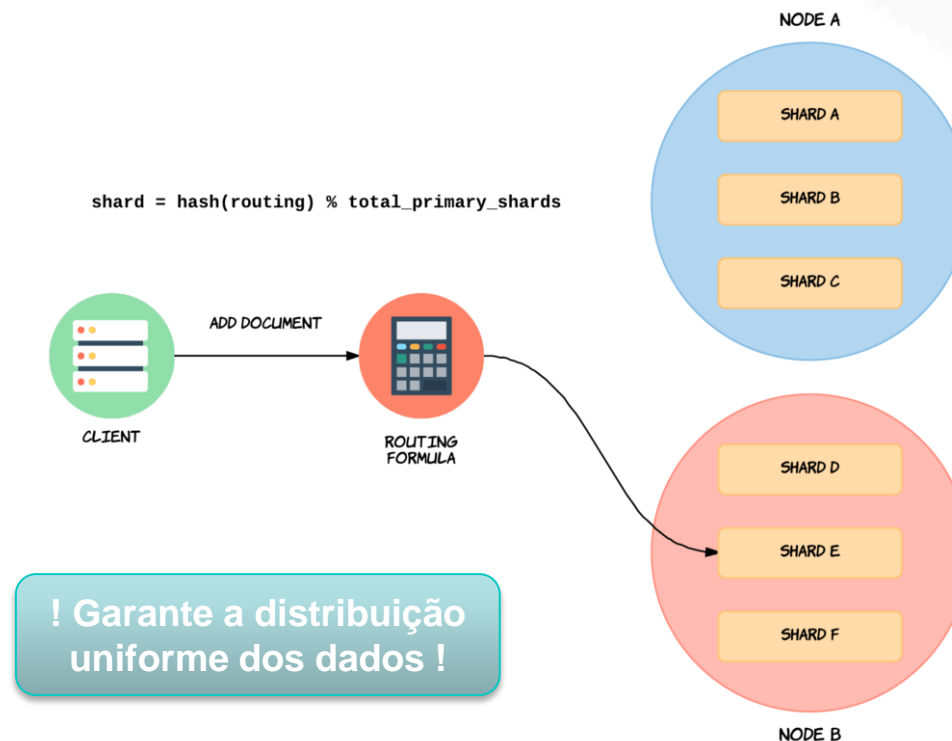


- **Sharding:**
 - Quantidade de shards especificada no momento da criação do índice.
 - Default de **5 shards**.
 - Os shards podem ser hospedados em qualquer nó dentro do cluster.



Elasticsearch - Arquitetura

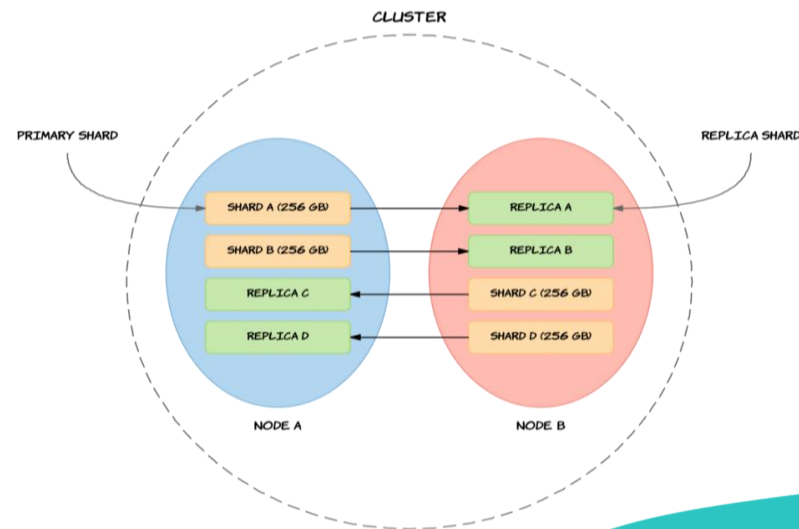
- **Sharding:**
 - **Roteamento (*routing*):**
para saber em qual shard armazenar um novo documento e como encontrá-lo, de forma transparente para os usuários.



Elasticsearch - Arquitetura



- **Replicação:**
 - Garantir a alta disponibilidade do ambiente e dos dados.
 - Tolerância a falhas.
 - Cópia de shards para outros nós:
 - **Fragmento primário (*primary shard*).**
 - Fragmentos replicados: **réplicas**.
 - **Grupo de replicação.**
 - Default: 1 réplica por shard.
 - 1 shard primário e 1 shard réplica.



Elasticsearch - Arquitetura

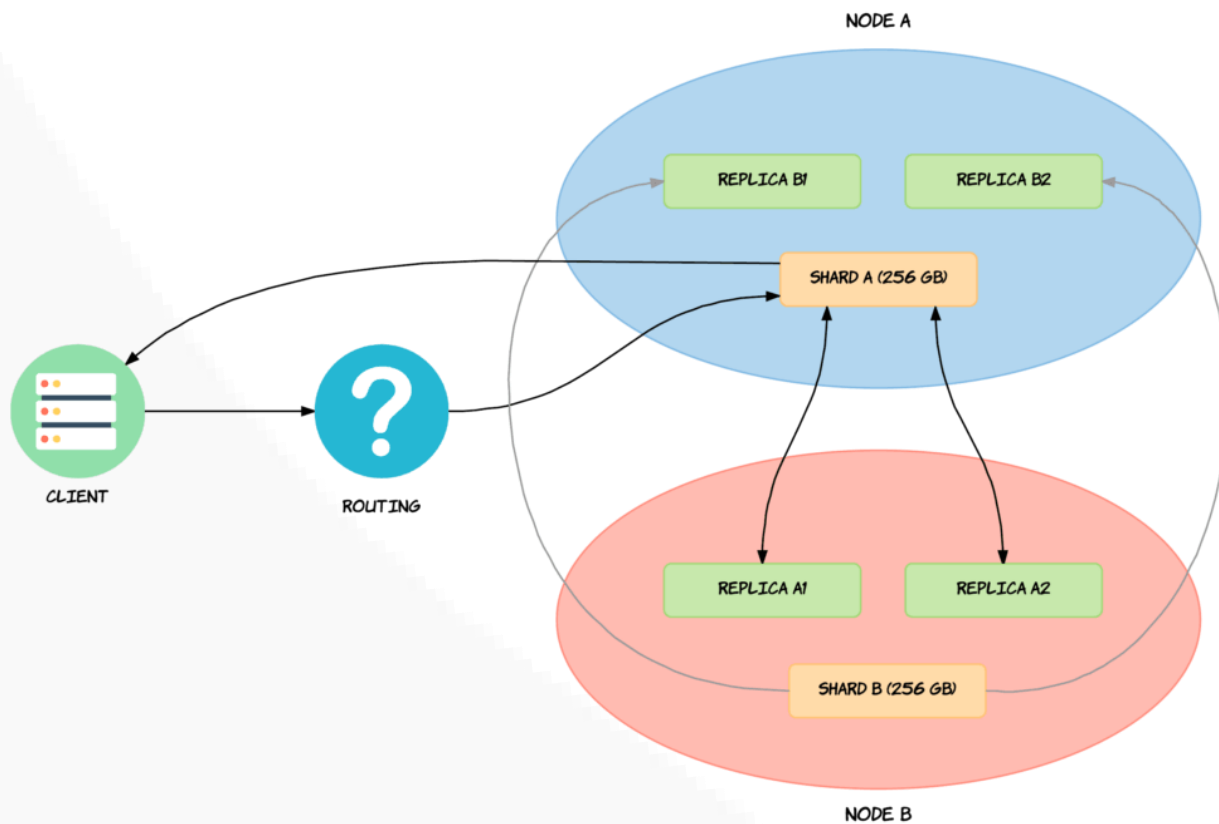


- **Replicação:**

- Topologia de replicação hierárquica: ***primary backup***.
- Operações que afetam o índice (adição, atualização ou remoção de documentos) são feitas apenas **no shard primário**.
- Shard primário é responsável por validar as operações e garantir que tudo esteja OK, estruturalmente e semanticamente falando.
- O shard primário, após validar e executar localmente a operação, encaminha para cada um dos shards uma réplica no grupo de replicação em questão.

Elasticsearch - Arquitetura

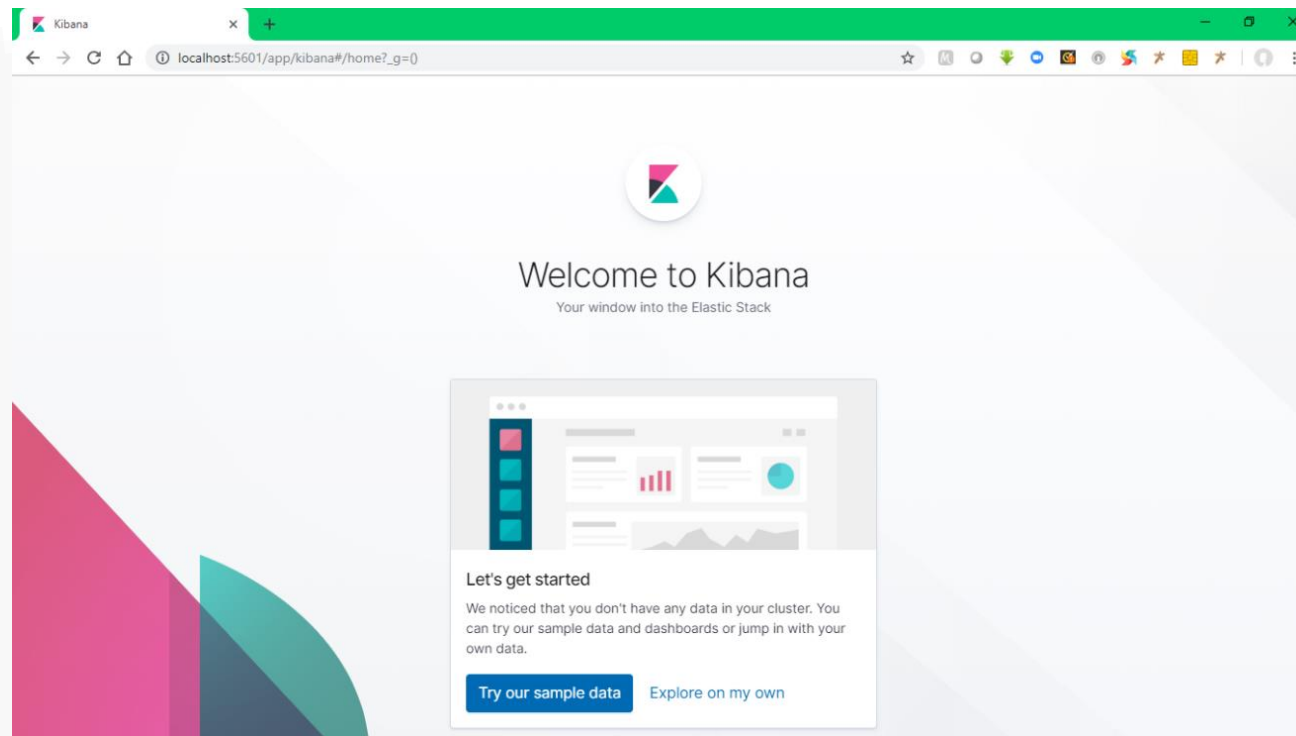
- Replicação síncrona:



Elasticsearch – Utilização



- Também pelo Kibana.

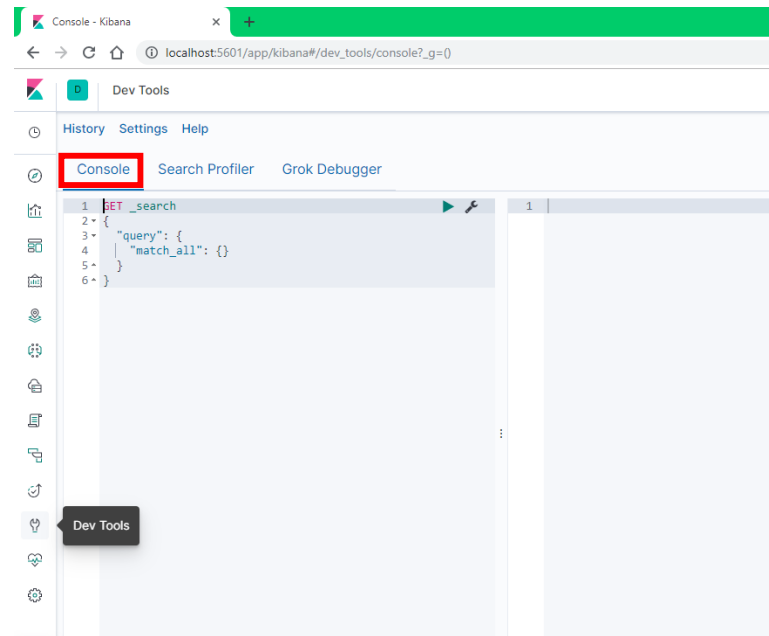


Elasticsearch – Utilização



- **Dev Tools:**

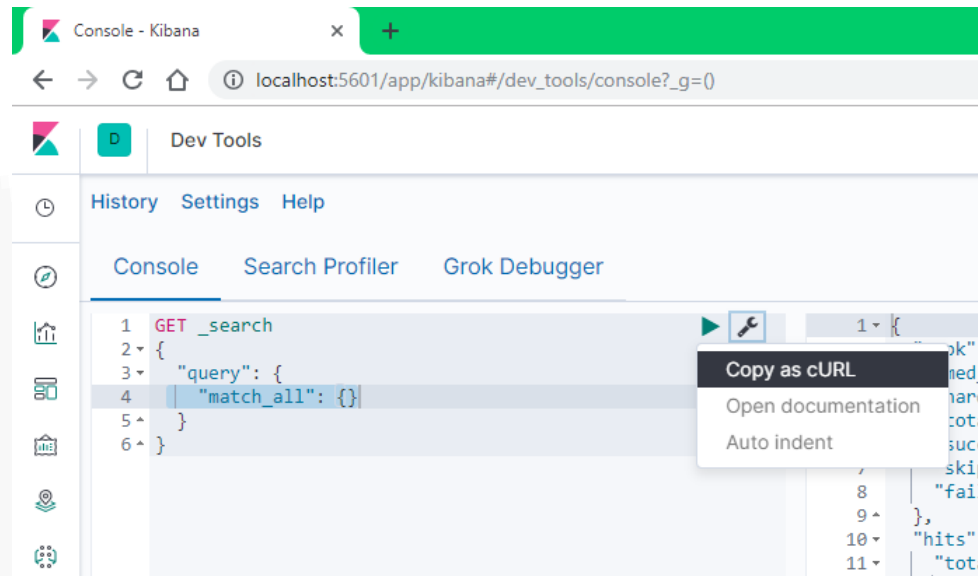
- Console dentro do Kibana.
- Criar / gerenciar índices.
- Inserir e manipular documentos.
- Sem preocupar com cabeçalhos, HTTP ou em formatar respostas.
- Envio das requisições HTTP:
 - Verbo HTTP (*GET / POST / PUT / DELETE*) +
 - Requisição URI + corpo (*body*) da requisição.



Elasticsearch – Utilização



- **Console** do **Dev Tools**:
 - Atalho para gerar o script no formato de requisições **cURL**.



Elasticsearch – Utilização

- *Para inserir documentos no índice: requisição **POST** para uma URI:*
 - Formada pelo nome do índice (não precisa estar criado previamente) +
 - Um *endpoint* (**_doc** ou **_create**) +
 - ID do documento (opcional, pois caso não seja informado, a API invocada gerará um automaticamente) +
 - Documento JSON com os dados do documento que deseja-se inserir.

Elasticsearch – Utilização



- Inserção de documento com ID explícito:

A screenshot of the Kibana Dev Tools console. The left pane shows a REST client request: a POST to /equipamentos/_doc/1 with a JSON body containing document details. The right pane shows the response, which is a 201 status code and a JSON object indicating the document was created successfully. The response includes fields like _index, _type, _id, _version, _result, _shards, _seq_no, and _primary_term.

```
1 POST /equipamentos/_doc/1
2 {
3   "nome": "NOTE001",
4   "usuario":
5   {
6     "primeironome": "Gustavo",
7     "sobrenome": "Aguilar"
8   },
9   "tipo": "NOTEBOOK"
10 }
11
12
13
```

```
1 {
2   "_index": "equipamentos",
3   "_type": "_doc",
4   "_id": "1",
5   "_version": 1,
6   "result": "created",
7   "shards": {
8     "total": 2,
9     "successful": 1,
10    "failed": 0
11  },
12   "_seq_no": 0,
13   "_primary_term": 1
14 }
```

Elasticsearch – Utilização



- Inserção de documento com ID automático.

A screenshot of the Kibana Console interface. The browser address bar shows 'localhost:5601/app/kibana#/dev_tools/console?_g=()'. The console is titled 'Dev Tools' and has tabs for 'History', 'Settings', and 'Help'. Below these are tabs for 'Console', 'Search Profiler', and 'Grok Debugger'. The 'Console' tab is active, showing a REST client interface. On the left, a POST request is defined for the endpoint '/equipamentos/_doc' with a JSON body: { "nome": "NOTE001", "usuario": { "primeironome": "Gustavo", "sobrenome": "Aguilar" }, "tipo": "NOTEBOOK" }. On the right, the response is displayed as a JSON object: { "_index": "equipamentos", "_type": "_doc", "_id": "CBFKhWoBFTkgddxyuuso", "_version": 1, "result": "created", "_shards": { "total": 2, "successful": 1, "failed": 0 }, "_seq_no": 1, "_primary_term": 1 }. The response is numbered 1 through 15 on the left margin. The console interface includes a left sidebar with various tool icons and a top navigation bar with a green header.

Elasticsearch – Utilização



- Pode-se usar também o verbo HTTP **PUT**:
 - De forma similar ao POST.
 - É preciso especificar, de forma explícita, o ID do documento.

A screenshot of the Kibana Dev Tools Console. The browser address bar shows 'localhost:5601/app/kibana#/dev_tools/console?_g=()'. The 'Dev Tools' panel is open, showing the 'Console' tab. A red box highlights the PUT request: 'PUT /equipamentos/_doc/2'. The request body is a JSON object: { 'nome': 'DESKTOP001', 'usuario': { 'primeironome': 'Juliana', 'sobrenome': 'Gonzaga' }, 'tipo': 'DESKTOP' }. The response is a JSON object: { '_index': 'equipamentos', '_type': '_doc', '_id': '2', '_version': 1, 'result': 'created', '_shards': { 'total': 2, 'successful': 1, 'failed': 0 }, '_seq_no': 2, '_primary_term': 1 }.

```
PUT /equipamentos/_doc/2
{
  "nome": "DESKTOP001",
  "usuario": {
    "primeironome": "Juliana",
    "sobrenome": "Gonzaga"
  },
  "tipo": "DESKTOP"
}
```

```
{
  "_index": "equipamentos",
  "_type": "_doc",
  "_id": "2",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 2,
  "_primary_term": 1
}
```

Elasticsearch – Utilização



- Consultar documentos pelo ID ➔ verbo HTTP **GET**.

A screenshot of the Kibana Dev Tools console. The browser address bar shows 'localhost:5601/app/kibana#/dev_tools/console?_g=()'. The console has tabs for 'History', 'Settings', and 'Help'. Below these are 'Console', 'Search Profiler', and 'Grok Debugger'. The 'Console' tab is active, showing a single command: '1 GET equipamentos/_doc/1'. The response is a JSON object: '{ "_index": "equipamentos", "_type": "_doc", "_id": "1", "_version": 1, "_seq_no": 0, "_primary_term": 1, "found": true, "_source": { "nome": "NOTE001", "usuario": { "primeironome": "Gustavo", "sobrenome": "Aguilar" }, "tipo": "NOTEBOOK" } }'. The response is displayed in a light blue background with line numbers 1 through 17 on the left and right sides of the JSON structure.

```
1 GET equipamentos/_doc/1
2 {
3   "_index": "equipamentos",
4   "_type": "_doc",
5   "_id": "1",
6   "_version": 1,
7   "_seq_no": 0,
8   "_primary_term": 1,
9   "found": true,
10  "_source": {
11    "nome": "NOTE001",
12    "usuario": {
13      "primeironome": "Gustavo",
14      "sobrenome": "Aguilar"
15    },
16    "tipo": "NOTEBOOK"
17  }
18 }
```

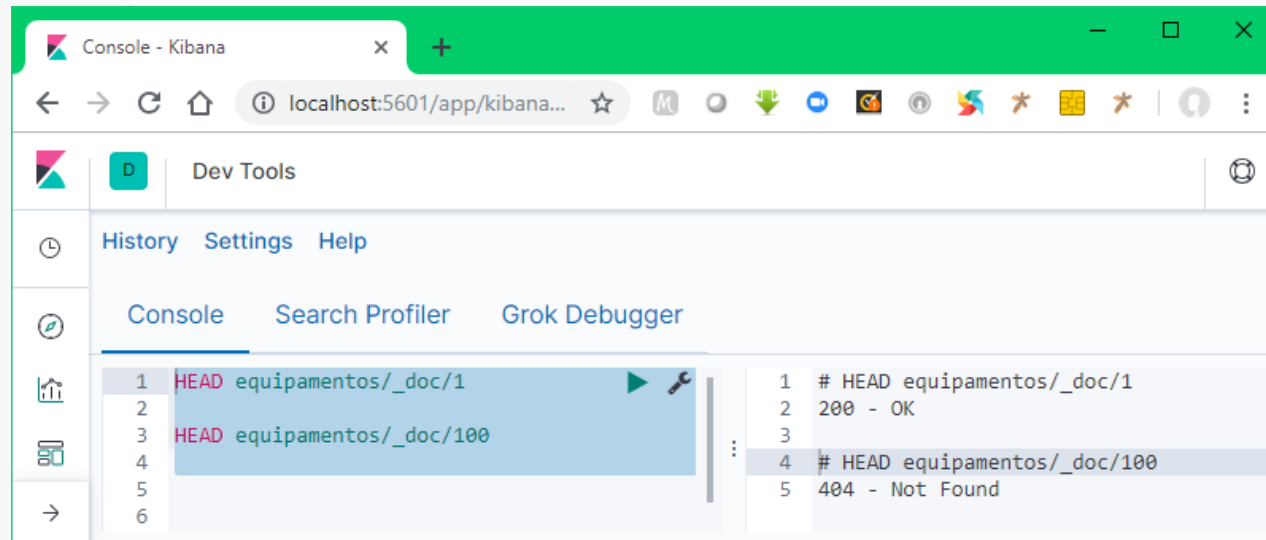
Elasticsearch – Utilização



- Consultar mais documentos → API **_mget** em conjunto com o **GET**:

GET equipamentos/_doc/_mget { "ids": ["1","2"] }

- Verificar existência de um documento pelo ID → verbo **HEAD**:



Elasticsearch – Utilização



- **Atualização (*update*):**

- Endpoint *_update*

- ***POST NOME_INDICE/_update/ID***

- {

- "doc":***

- { ATUALIZACAO_AQUI }***

- }

- Mantém os outros campos.

```
1 POST /equipamentos/_update/2
2 {
3   "doc": { "nome": "DESKTOP002",
4           "fornecedor": "DELL"
5   }
6 }
7
8 GET equipamentos/_doc/2
9
10
11
```

```
1 # POST /equipamentos/_update/2
2 {
3   "_index" : "equipamentos",
4   "_type" : "_doc",
5   "_id" : "2",
6   "_version" : 2,
7   "result" : "updated",
8   "_shards" : {
9     "total" : 2,
10    "successful" : 1,
11    "failed" : 0
12  },
13   "_seq_no" : 9,
14   "_primary_term" : 1
15 }
16
17
18 # GET equipamentos/_doc/2
19 {
20   "_index" : "equipamentos",
21   "_type" : "_doc",
22   "_id" : "2",
23   "_version" : 2,
24   "_seq_no" : 9,
25   "_primary_term" : 1,
26   "found" : true,
27   "_source" : {
28     "nome" : "DESKTOP002",
29     "usuario" : {
30       "primeironome" : "Juliana",
31       "sobrenome" : "Gonzaga"
32     },
33     "tipo" : "DESKTOP",
34     "fornecedor" : "DELL"
35   }
36 }
```

Elasticsearch – Utilização



- Deletar apenas um documento:
 - Verbo HTTP ***DELETE***.

```
1 DELETE postagens/_doc/1
2
3 GET postagens/_doc/1
4
5
```

```
1 # DELETE postagens/_doc/1
2 {
3   "_index" : "postagens",
4   "_type" : "_doc",
5   "_id" : "1",
6   "_version" : 6,
7   "result" : "not_found",
8   "_shards" : {
9     "total" : 2,
10    "successful" : 1,
11    "failed" : 0
12  },
13   "_seq_no" : 5,
14   "_primary_term" : 1
15 }
16
17
18 # GET postagens/_doc/1
19 {
20   "_index" : "postagens",
21   "_type" : "_doc",
22   "_id" : "1",
23   "found" : false
24 }
25
```

Elasticsearch – Utilização



- Deletar um conjunto de documentos:


- Verbo HTTP **POST**;
- API **_delete_by_query**;
- Passar a variável **query**
 - Será o filtro (condição *where*).

```
POST equipamentos/_delete_by_query
{
  "query": {
    "match": {"tipo": "DESKTOP"}
  }
}
```


```
POST NOME_INDICE/_delete_by_query
{
  "query": {
    "match": {"campo": "valor"}
  }
}
```


Elastic Cloud



 Elastic

Cloud

 **Elasticsearch Service**




Get started with Elasticsearch Service

Create your first deployment to manage an Elasticsearch cluster on the cloud platform of your choice.
Add additional Elastic products to your deployment like Kibana, machine learning, or APM.

[Start your free trial](#)

Platform features


- ✓ Cloud hosting on AWS, GCP or Azure
- ✓ Logs, metrics, and APM in one place
- ✓ Includes machine learning, security, and more
- ✓ One-click upgrades with no downtime
- ✓ Same-day new version releases
- ✓ Monitored 24/7

 **News**

[Elastic Stack 7.10.2 released](#)
JANUARY 14, 2021 **New!**

[Elastic Cloud Terraform provider now available in beta](#)
DECEMBER 17, 2020 **New!**

[Elastic Stack 7.10.1 released](#)
DECEMBER 9, 2020 **New!**

 **Training**

Get started with our free training

Build essential skills and learn Elastic with free in-training in the Elastic Learning Portal

[Access free courses](#)

<https://cloud.elastic.co>

Dúvidas do Trabalho Prático

IGTi





Quiz Premiado



- Vale-presente (gift card) da **Americanas.com**, no valor de **R\$25,00**, para o **primeiro colocado**.
- Acesse o site **www.kahoot.it** ou o app **Kahoot!**
- Digite o Game PIN: **??????**

Kahoot!