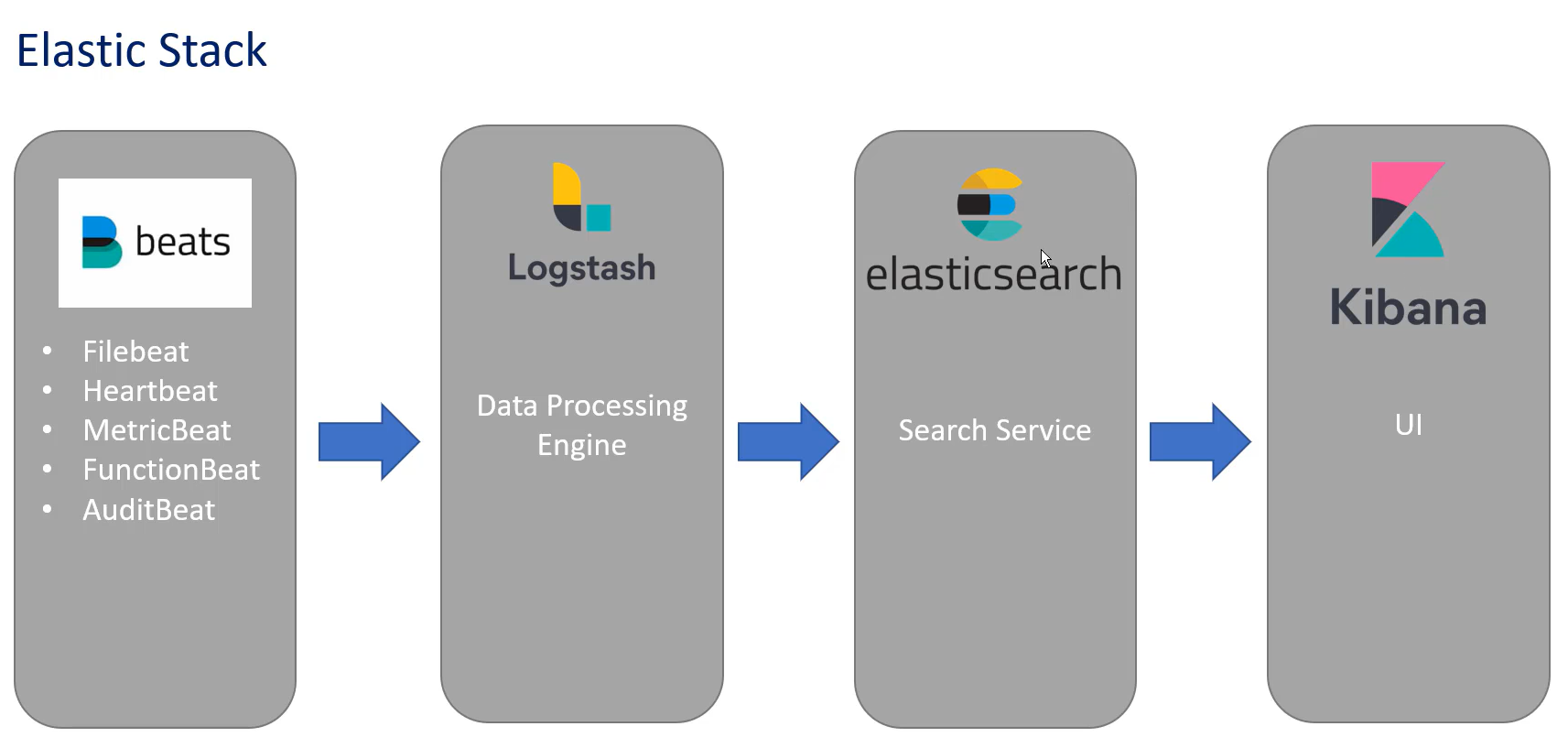
**Elastic Search Nedir ?**

* Open Source olan bir arama motorudur. Java ile geliştirilmiştir.
* Full text Search
* Aynı zamanda No Sql veritabanıdır. Json formatında verileri kaydeder.
* Diğer No Sql veritabanlarından farkı aynı zamanda arama motoru olmasıdır
* Hacimli datalarda arama yapılmak istendiği zaman en uygunudur.
* Elastic Search herşeyi bütün kelimeleri index liyor. Bu yüzden hızlıdır.
* Structured & Unstructered veriler üzerinde arama yapar.

Structured => Normal veritabanınada yapmış olduğumuz aramlardır

Unstructered => text üzerinde yapılan aramalrdır

* En Önemli özelliği unstructured datalar üzerinde hızlıca arama yapabilmesidir



**Elastic Search Docker Kurulumu**

* **Docker Desktop yoksa kurulmalıdır.**
* **Docker Hub üye olunmalıdır. Dünyadaki tüm image ların depolandığı yerdir.**

**metin, ekran görüntüsü, yazı tipi, meneviş mavisi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Term Query =>** Tam bir veri ye göre arama yapar. ProductId, UserName gibi. Tam veri üzerinde arama yapılacaksa **keyword** tipinde tanımlanmalıdır. (**Örnek 1**) Aşağıdaki örnekte direk “Sonya” olarak arama yapıyor. Büyük harf küçük harf duyarlılığı vardır. (**Örnek 2**) büyük küçük harfe duyarlı değildir

**Örnek 1 ;**

**metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Örnek 2 ;**

**metin, yazı tipi, çizgi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Örnek 3; (iki farklı isimi arama)**

**metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Aşağıdaki **customer\_first\_name** property si hem text, hem de keyword olarak tanımlandığı için arama yapılırken şu mantığa göre yapılır;

**“Büyük Mavi Kalem”**

->ilk tipi text olduğu için inverted table lara parçalanarak yazılır

büyük,mavi,kalem

**->**ikinci tipi keyword olduğu için inverted table a bütün olarak yazılır

Büyük Mavi Kalem

**Not :** Böylece arama yapılırken her bir kelimeyi tek tek yazarakta arama yapıalbilir. Bütün yazılarak ta arama yapılabilir

**metin, yazı tipi, makbuz, beyaz içeren bir resim

Açıklama otomatik olarak oluşturuldu**

metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, iş kartı içeren bir resim

Açıklama otomatik olarak oluşturuldu

**NOT :**

* ELASTİC VERSİON 7 YE KADAR OLANLARDA **NEST** KÜTÜPHANESİ
* VERSİON 8 OLANLARDA İSE **ELASTICSEARCH** KÜTÜPHANESİ KULLANILMALIDIR
* SIFIRDAN ELASTİC KURULDU İSE VERSİON 8 DİR VE **ELASTICSEARCH** KÜTÜPHANESİ KULLANILMALIDIR
* AMA **NEST** KÜTÜPHANESİ VERSİON 8 İLE DE UYUMLUDUR, KULLANILABİLİR

1. **Tokenization:**
   * **Tanım:** Tokenization, bir metni daha küçük parçalara, yani "token"lara ayırma sürecidir. Bir token, metindeki anlamlı bir birimi temsil eder. Bu bir kelime, bir cümle veya bir noktalama işareti olabilir.
   * **Örnek:** "Merhaba, dünya!" cümlesi tokenlara ayrıldığında, örnek tokenlar şu şekilde olabilir: ["Merhaba", ",", "dünya", "!"]
2. **Normalization:**
   * **Tanım:** Normalization, metin verilerini standart bir formata getirme sürecidir. Bu, metin verilerindeki çeşitli değişiklikleri düzenlemeyi içerir, böylece benzer anlamlı ifadeler aynı temsil ediliyor ve işleme daha uygun hale geliyor.
   * **Örnek:** Normalization, metin verilerindeki büyük küçük harf farklarını giderme, özel karakterleri temizleme, kelime köklerini bulma (stemming) veya eş anlamlı kelimeleri birleştirme (lemmatization) gibi işlemleri içerebilir.

**Örnek Uygulama:** Bir metin üzerinde tokenization ve normalization uygulamak için şu adımları takip edebiliriz:

Metin: "Bu bir örnek cümledir; cümlenin içinde kelimeler ve noktalama işaretleri bulunmaktadır."

1. **Tokenization:**
   * ["Bu", "bir", "örnek", "cümledir", ";", "cümlenin", "içinde", "kelimeler", "ve", "noktalama", "işaretleri", "bulunmaktadır", "."]
2. **Normalization:**
   * Küçük harfe dönüştürme: ["bu", "bir", "örnek", "cümledir", ";", "cümlenin", "içinde", "kelimeler", "ve", "noktalama", "işaretleri", "bulunmaktadır", "."]
   * Noktalama işaretlerini temizleme: ["bu", "bir", "örnek", "cümledir", "cümlenin", "içinde", "kelimeler", "ve", "noktalama", "işaretleri", "bulunmaktadır"]
   * Lemmatization (kelime köklerini bulma): ["bu", "bir", "örnek", "cümle", "cümle", "iç", "kelime", "ve", "noktalama", "işaret", "bulunmak"]

**YENİ INDEX OLUŞTURMA**

**metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

* Tip **text** olarak tanımlanırsa analiz (indexleme, Normalization, Tokenization) sürecinden geçer. Çünkü full text search yapılır. Örnek olarak bi makale aranacağı zaman, makalenin tamamı yazılmaz. Indexleme yapıldığı için yazılan bir kelimeye göre makaleyi getirebilir
* Tip **keyword** ise analiz sürecinden geçmez direk arama yapmak içindir. Örneke olarak userid, tckimlik, telefonno gibi analiz edilmemesi, parçalanmaması gereken veriler keyword olarak tanımlanmalıdır

metin, yazılım, web sayfası, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu