Fakulta informatiky a informačných technológií STU v Bratislave

**Zadanie 5 – Elastic Search**

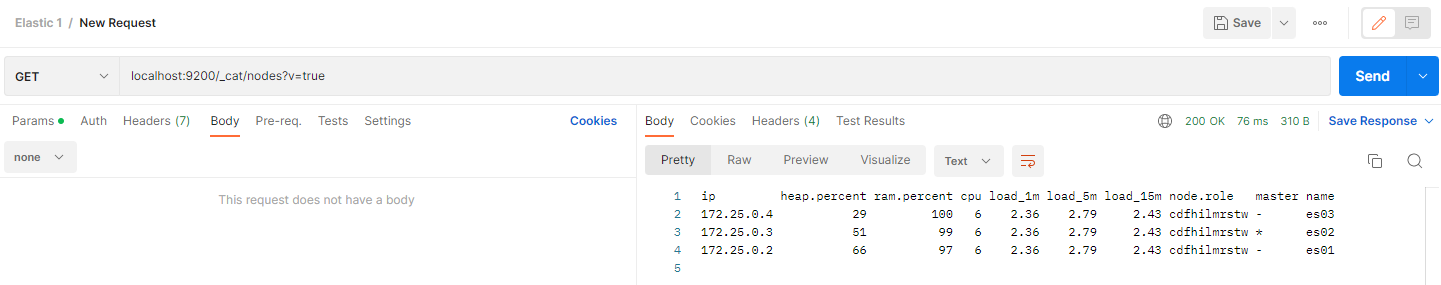
**29.9.2022**

**Matej Delinčák**

**Github classroom:** <https://github.com/FIIT-DBS/zadanie-pdt-mateju25>

**1. Rozbehajte si 3 inštancie Elasticsearch-u**

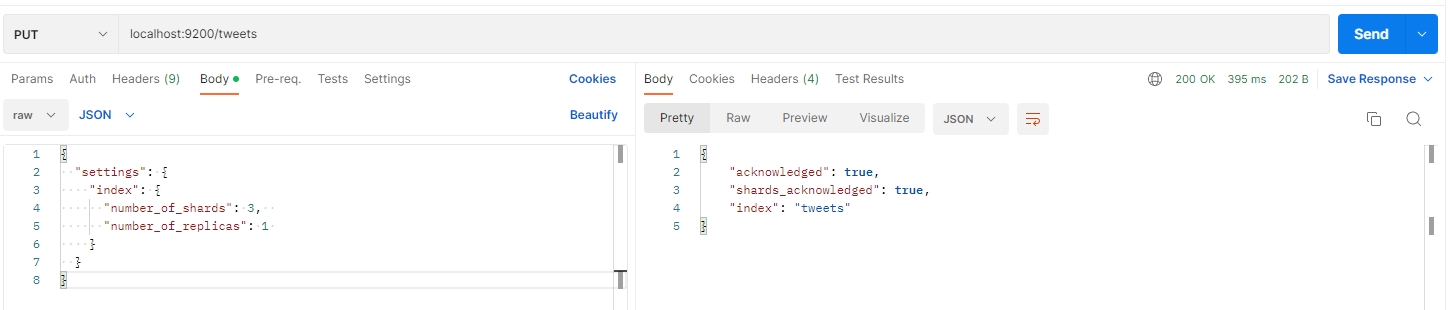
Pomocou docker compose som si rozbehal 3 inštancie elasticu. Compose yml bude priložený pri dokumentácií. Vytvoril som teda 3 inštancie do jedného clustru a jeden z nich označil ako master.



**2. Vytvorte index pre Tweety, ktorý bude mať “optimálny“ počet shardov a replík pre 3**

**nódy (aby tam bola distribúcia dotazov vo vyhľadávaní, aj distribúcia uložených dát)**

V dokumentácií som sa dočítal, že tri shardy je ideálne na 3 nody ak nemám veľa pamäte. Tak to budem mať aj paralelne spracovávanie. Počet replík som nastavil na 1, ktorá je aj defaultná hodnota. Počet replík je dôležitý pre dostupnosť systému. Ale nemyslím, že u mňa systém bude tak zaťažený (aby sa stala nejaká nehoda), aby som potreboval až dve repliky.



Overenie:

Obrázok, na ktorom je text

Automaticky generovaný popis

**3. Vytvorte mapping pre normalizované dáta z Postgresu (denormalizujte ich) – Každý**

**Tweet teda musí obsahovať údaje rovnaké ako máte už uložené v PostgreSQL (všetky**

**tabuľky). Dbajte na to, aby ste vytvorili polia v správnom dátovom type (polia ktoré**

**má zmysel analyzovať analyzujte správne, tie ktoré nemá, aby neboli zbytočne**

**analyzované (keyword analyzer)) tak aby index nebol zbytočne veľký, pozor na**

**nested – treba ho použiť správne. Mapovanie musí byť striktné. Čo sa týka väzieb cez**

**referencies – pre ne zaindexujte type vsťahu, id, autor (id, name, username), content**

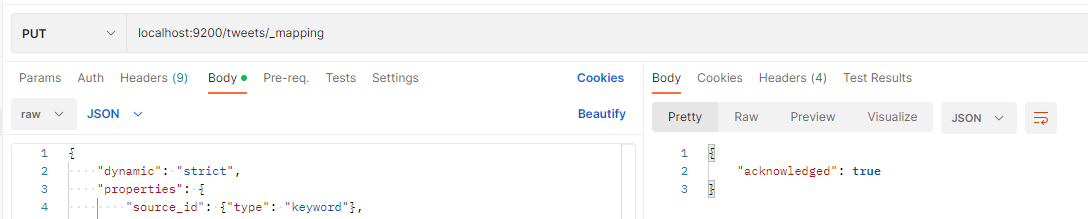
**a hashtags.**

Vytvoril som nasledovné mapovanie. Pre čísla som použil integer, pre string typ text. Typ keyword som použil pre id stĺpce a hastagy, kedže v dokumentácií som sa dočítal, že sa najčastejšie používajú na tento typ. Typ nested som použil pre author, content\_domains a context\_entities. Z dokumentácie: *„The nested type is a specialised version of the object data type that allows arrays of objects to be indexed in a way that they can be queried independently of each other.“*

Obrázok, na ktorom je text

Automaticky generovaný popisObrázok, na ktorom je stôl

Automaticky generovaný popis



**4. Pre index tweets vytvorte 3 vlastné analyzéry (v settings) nasledovne:**

**a. Analyzér "englando". Tento analyzér bude obsahovať nasledovné:**

**i. fitre: english\_possessive\_stemmer, lowercase, english\_stop,**

**english\_stemmer,**

**ii. char\_filter: html\_strip**

**iii. tokenizer: štandardný - ukážku nájdete na stránke elastic.co pre**

**anglický analyzér**

Vytvoril som nasledovný analyzér podľa zadania. Išlo to ľahko podľa dokumentácie. Všetky filtre okrem lowercase som musel ešte dotatočne customizovať, lebo taký názov neexistoval.

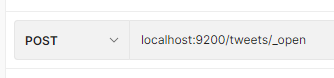
Obrázok, na ktorom je text

Automaticky generovaný popis

Ale predtým, než som mohol daný analyzér vytvoriť, tak som musel najprv vypnúť index.

Obrázok, na ktorom je text

Automaticky generovaný popis



**b. Analyzér custom\_ngram:**

**i. filtre: lowercase, asciifolding, filter\_ngrams (definujte si ho sami na**

**rozmedzie 1- 10)**

**ii. char\_filter: html\_strip**

**iii. tokenizer: štandardný**

Vytvorenie analyzéra custom\_ngram bolo taktiež priamočiare. Ale keď som vytváral filter\_ngrams, tak som dostal error hlášku, že maximálna hodnota ngramu pre index je 9. Tak som ešte musel nastaviť max. prístupnú hodnotu na 10.

**Obrázok, na ktorom je text

Automaticky generovaný popis**

**c. Analyzér custom\_shingles:**

**i. filtre: lowercase, asciifolding, filter\_shingles (definujte si ho sami a**

**dajte token\_separator: “”)**

**ii. char\_filter: html\_strip**

**iii. tokenizer: štandardný**

Pri vytvorení analyzéra custom\_shingles bolo potrebné vytvoriť custom filter pre typ shingle s parametrom token\_separator.

Obrázok, na ktorom je text

Automaticky generovaný popis

**d. Do mapovania pridajte:**

**i. každý anglický text (rátajme že každý tweet a description u autora je**

**primárne v angličtine) nech je analyzovaný novým analyzérom**

**"englando"**

**ii. Priraďte analýzery**

**1. a. author.name nech má aj mapovania pre custom\_ngram, a**

**custom\_shingles**

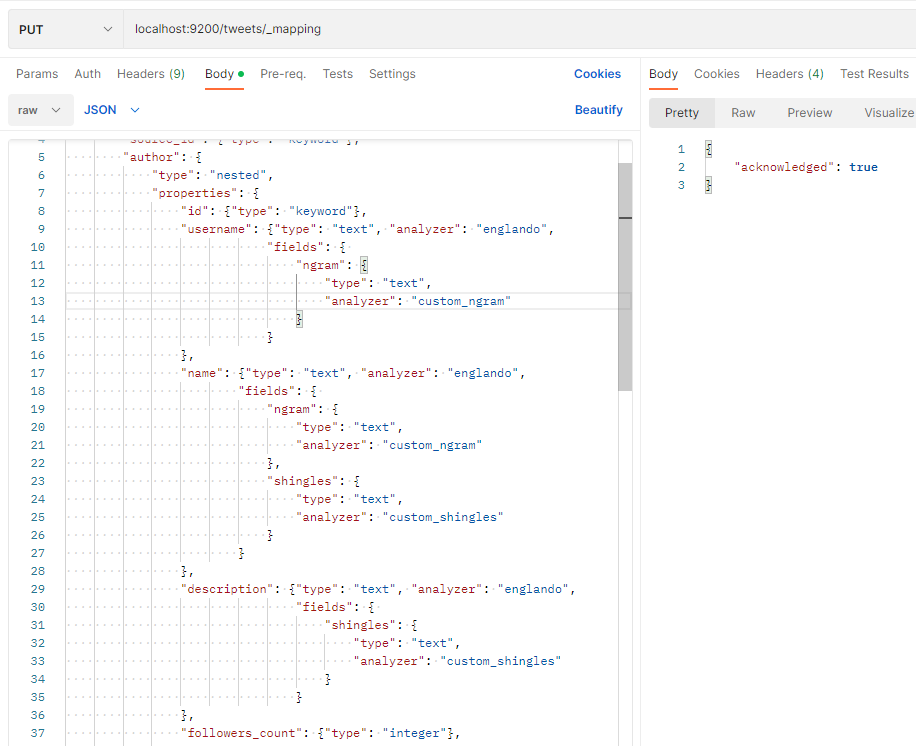
**2. b. author. screen\_name nech má aj custom\_ngram,**

**3. c. author.description nech má aj custom\_shingles. Toto platí aj**

**pre mentions, ak tam tie záznamy máte.**

**iii. Hashtagy indexujte ako lowercase**

Pre zadefinovanie viacero analyzátorov, som musel použiť fields parameter. Aby sa hashtagy indexovali ako lowercase, tak som si vytvoril normalizér s menom custom\_normalizer, ktorý ma filter lowercase v sebe.





**5. Vytvorte bulk import pre vaše normalizované Tweety.**

**6. Importujete dáta do Elasticsearchu prvych 5000 tweetov**

V súbore export.txt sa nachádza query podľa, ktorej som si vytiahol json pre každú konverzáciu. Tento json obsahuje donormalizované dáta z databázy. Pri query som využil funckie ako jsonb\_build\_object, ktorá vyskladala z dát json. Ďalej json\_agg, čo je window funkcia na agregáciu jsonov do poľa. Funkcia row\_to\_json vyvtvorí s viacerých stĺpcov jeden stĺpec, v ktorom je json. Novou vecou pre mňa bola funkcionalita LATERAL join, čo je v podstate niečo ako for cyklus. Túto novinku mi poradil cvičiaci, bez nej by som nevedel vytvoriť sql tak jednoducho. Ďalej pri importe do elasticu našiel nejaké znaky, čo robili problémy, tak som pomocou regexu takéto opravil.

Následne som si vytvoril python script, ktorý pomocou bulk api elasticu vložil 5000 tweetov. Overil som si import pomocou vytiahnutia všetkých dát z elasticu. A boli tam už všetky.

Obrázok, na ktorom je text

Automaticky generovaný popis

**7. Experimentujte s nódami, a zistite koľko nódov musí bežať (a ktoré) aby vám Elasticsearch vedel pridávať dokumenty, mazať dokumenty, prezerať dokumenty a vyhľadávať nad nimi? Dá sa nastaviť Elastic tak, aby mu stačil jeden nód? Čo je dôvodom toho že existuje nejaké kvórum?**