Fakulta informatiky a informačných technológií STU v Bratislave

Zadanie 5 – Elastic Search

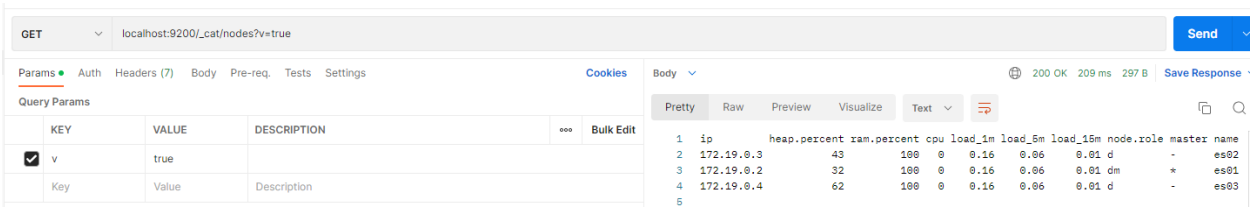
10.12.2022

Matej Delinčák

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

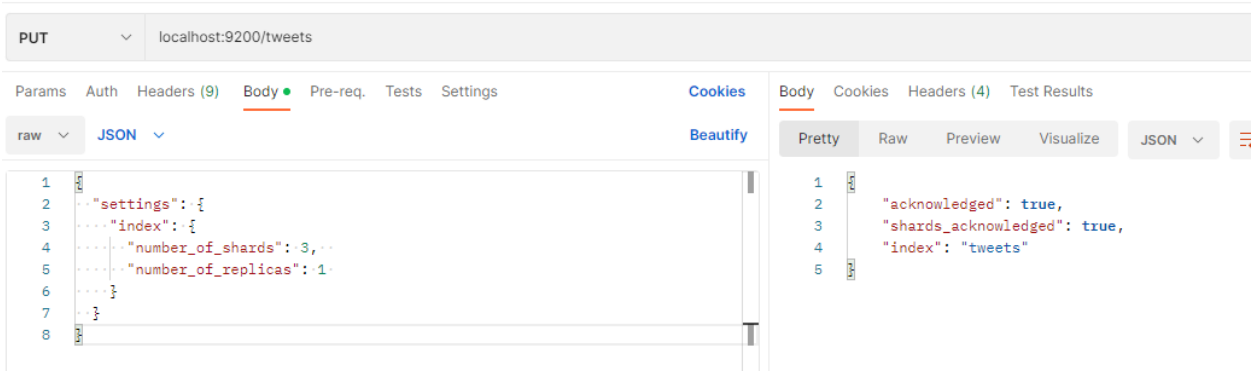
1. Rozbehajte si 3 inštancie Elasticsearch-u

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

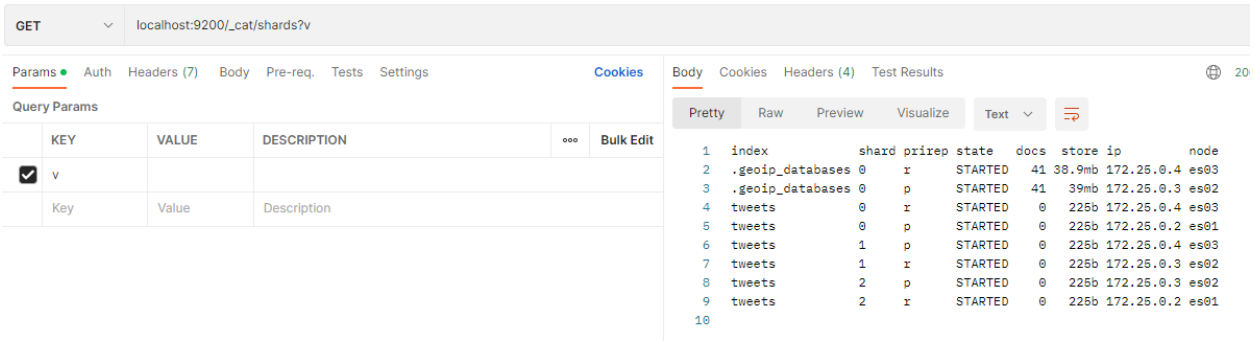


2. Vytvorte index pre Tweety, ktorý bude mať "optimálny" počet shardov a replík pre 3 nódy (aby tam bola distribúcia dotazov vo vyhľadávaní, aj distribúcia uložených dát)

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:



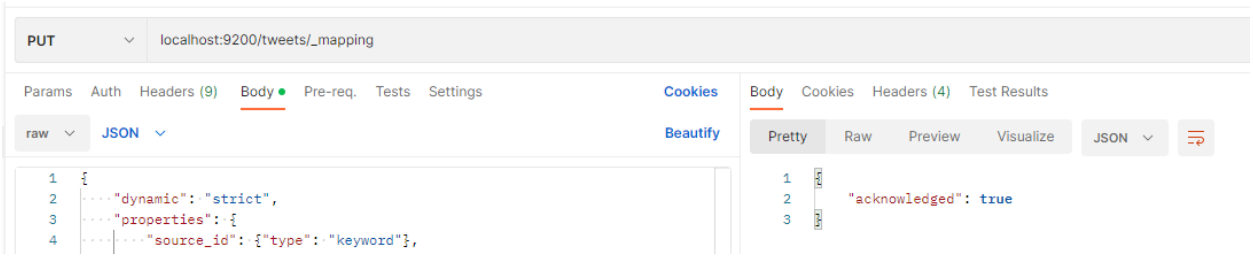
3. Vytvorte mapping pre normalizované dáta z Postgresu (denormalizujte ich) – Každý Tweet teda musí obsahovať údaje rovnaké ako máte už uložené v PostgreSQL (všetky tabuľky). Dbajte na to, aby ste vytvorili polia v správnom dátovom type (polia ktoré má zmysel analyzovať analyzujte správne, tie ktoré 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.“* Mapping bude priložený pri dokumentácií.

```
{
  "dynamic": "strict",
  "properties": {
    "source_id": {"type": "keyword"},
    "author": {
      "type": "nested",
      "properties": {
        "id": {"type": "keyword"},
        "username": {"type": "text"},
        "name": {"type": "text"},
        "description": {"type": "text"},
        "followers_count": {"type": "integer"},
        "following_count": {"type": "integer"},
        "listed_count": {"type": "integer"},
        "tweet_count": {"type": "integer"}
      }
    },
    "content": {"type": "text"},
    "language": {"type": "text"},
    "source": {"type": "text"},
    "retweet_count": {"type": "integer"},
    "reply_count": {"type": "integer"},
    "like_count": {"type": "integer"},
    "quote_count": {"type": "integer"},
    "possibly_sensitive": {"type": "boolean"},
    "created_at": {"type": "date"},
    "hashtags": {"type": "keyword"},
    "links": {
      "properties": {
        "title": {"type": "text"},
        "description": {"type": "text"},
        "url": {"type": "text"}
      }
    },
```

```
    "annotations": {
      "properties": {
        "value": {"type": "text"},
        "type": {"type": "text"},
        "probability": {"type": "double"}
      }
    },
    "context_annotations": {
      "type": "nested",
      "properties": {
        "domain": {
          "type": "nested",
          "properties": {
            "name": {"type": "text"},
            "description": {"type": "text"}
          }
        },
        "entity": {
          "type": "nested",
          "properties": {
            "name": {"type": "text"},
            "description": {"type": "text"}
          }
        }
      }
    },
```

```
    "conversation_references":{
      "properties":{
        "id" : {"type":"keyword"},
        "type": {"type": "text"},
        "hashtags": {"type": "keyword"},
        "content": {"type":"text"},
        "author": {
        "properties":{
          "id": {"type":"keyword"},
          "name": {"type":"text"},
          "username": {"type": "text"}
        }
        }
      }
```



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

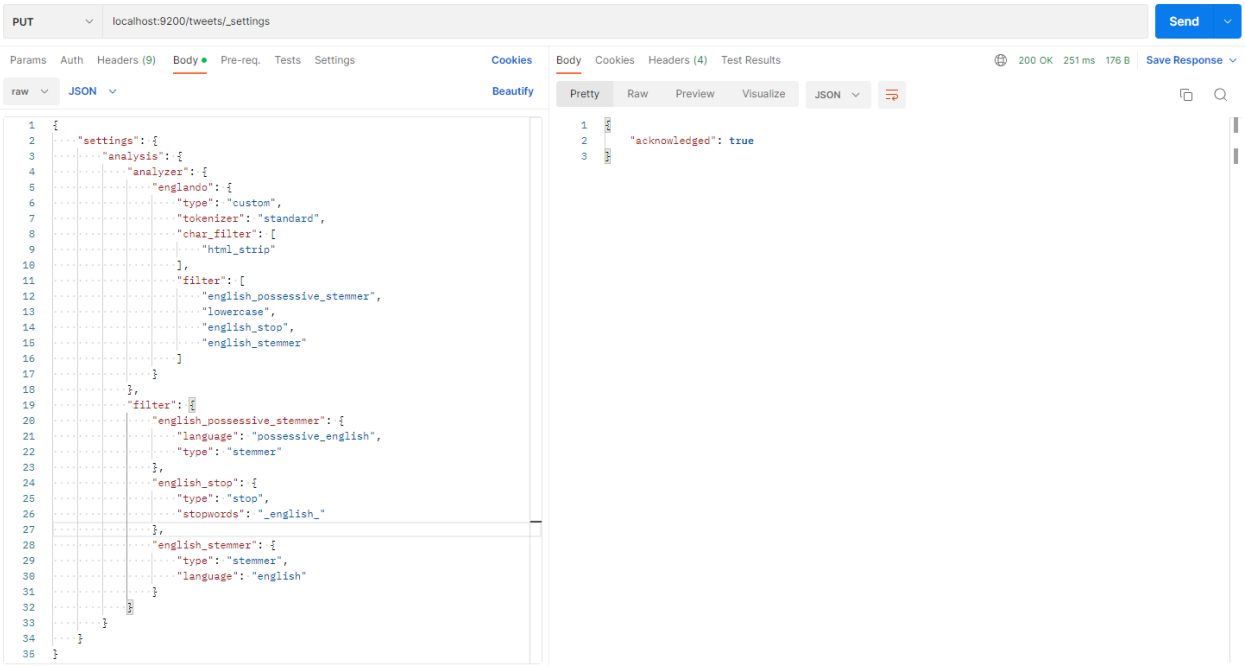
a. Analyzér "englando". Tento analyzér bude obsahovať nasledovné:

i. fitre: english_possessive_stemmer, lowercase, english_stop, english_stemmer,

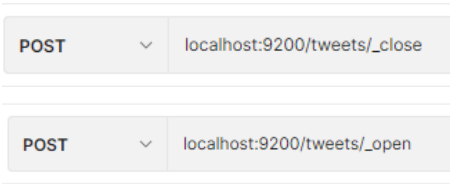
ii. char_filter: html_strip

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.



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



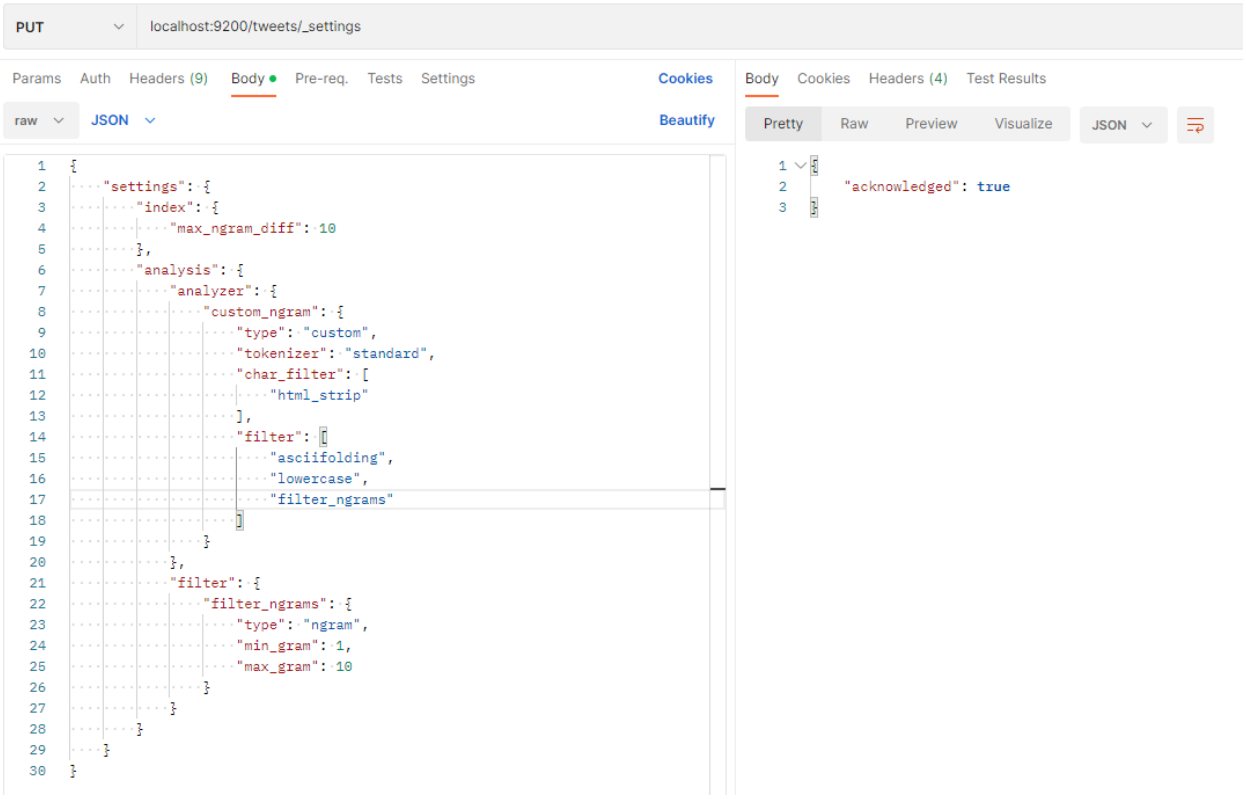
b. Analyzér custom_ngram:

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.



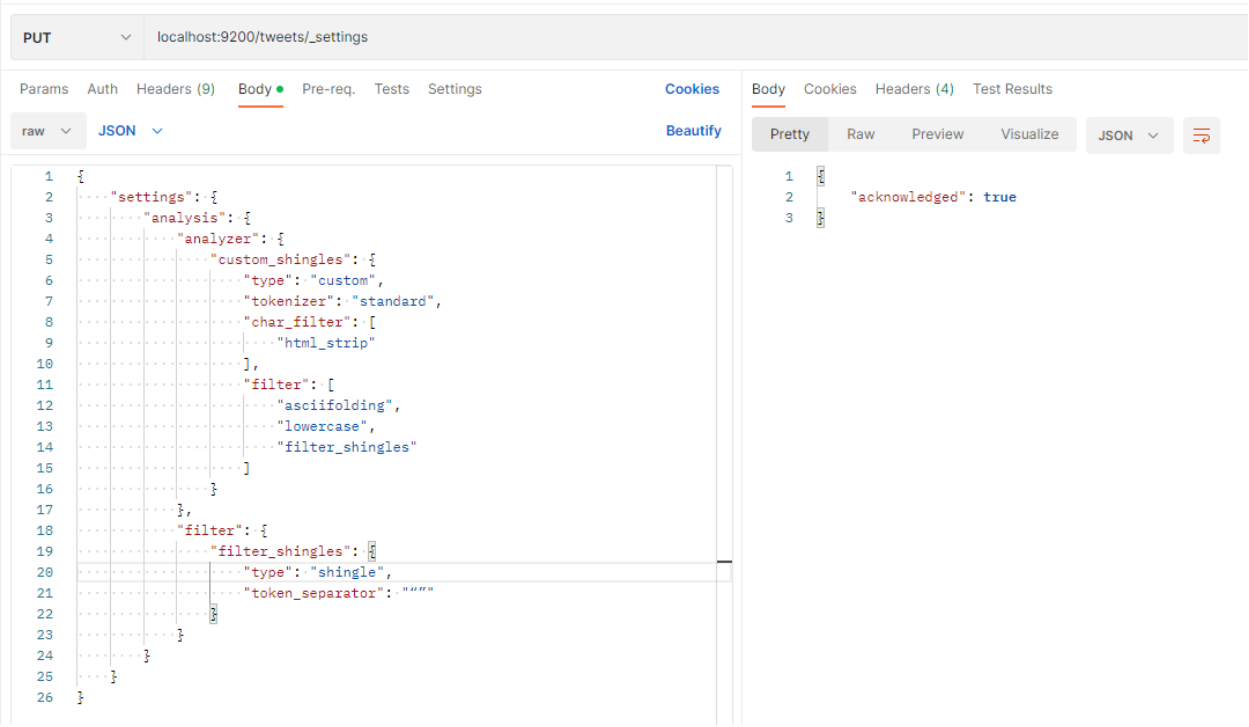
c. Analyzér custom_shingles:

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.



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

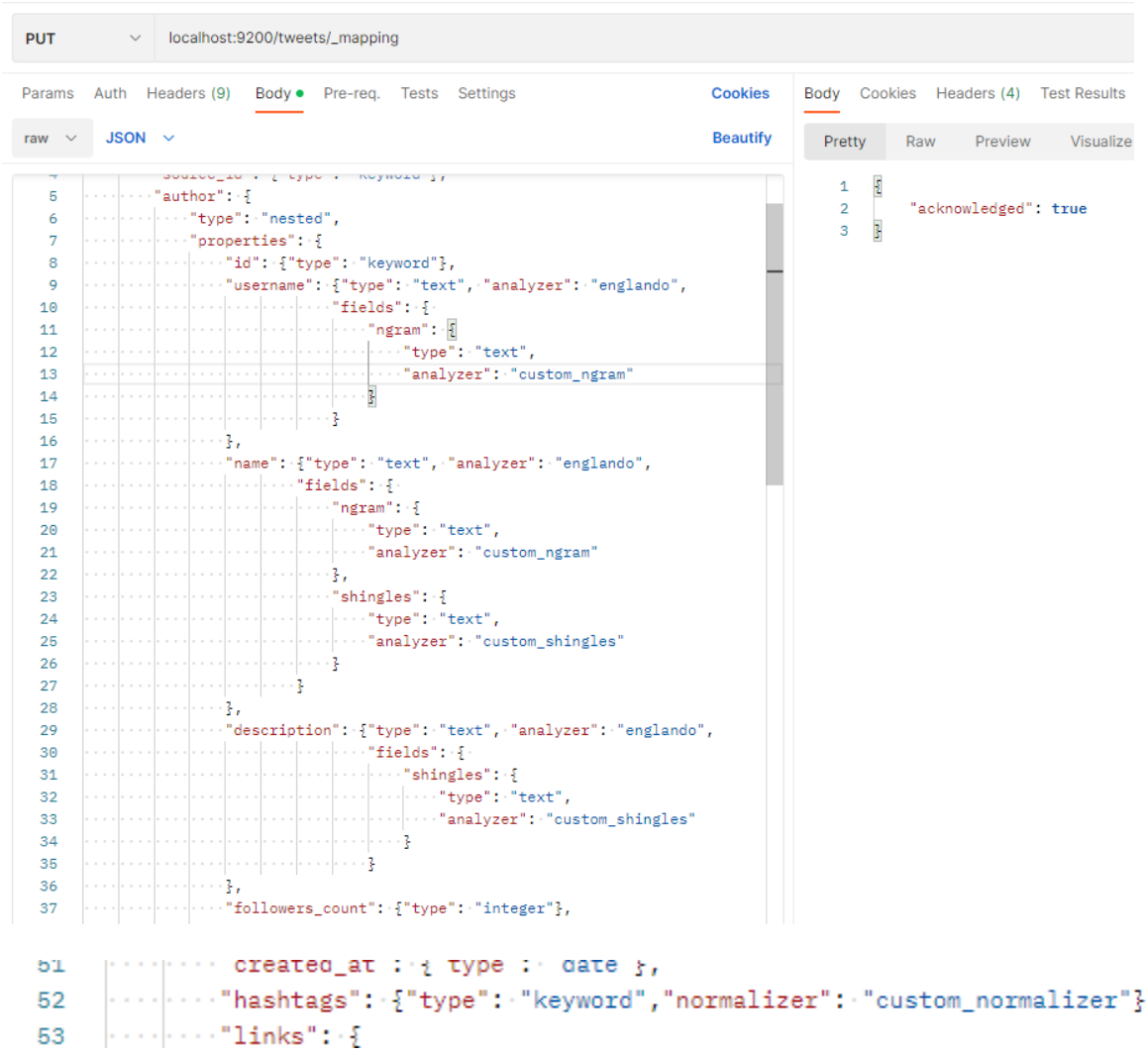
1. a. author.name nech má aj mapovania pre custom_ngram, a custom_shingles

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.

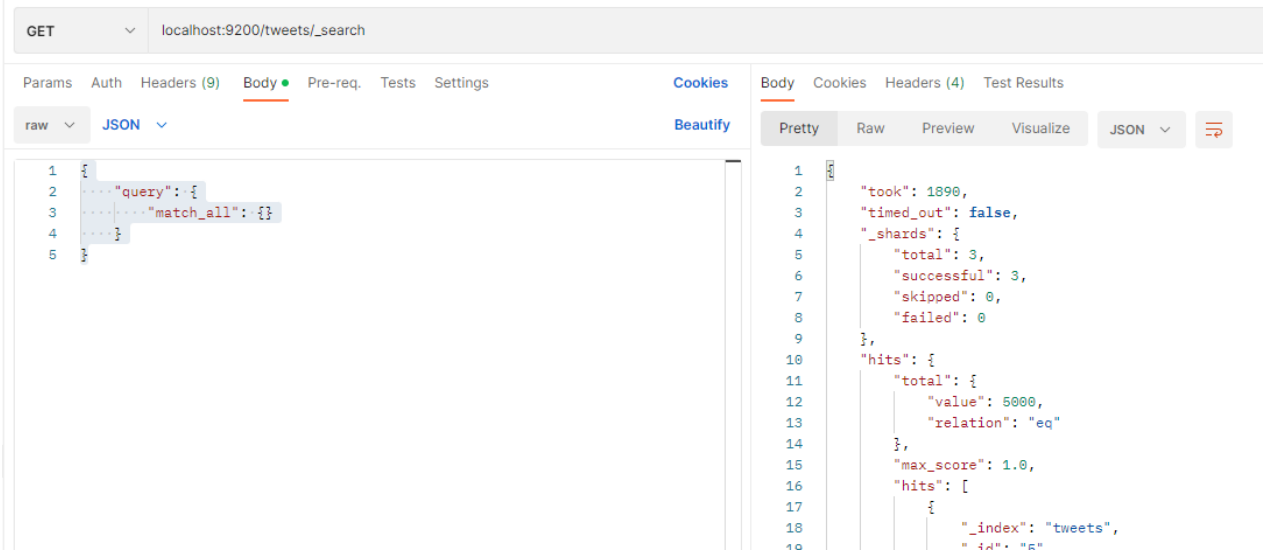



```
51          created_at : { type : date },
52        "hashtags": {"type": "keyword","normalizer": "custom_normalizer"},
53        "links": {
```

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. Export mi trval 13 hodín.

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



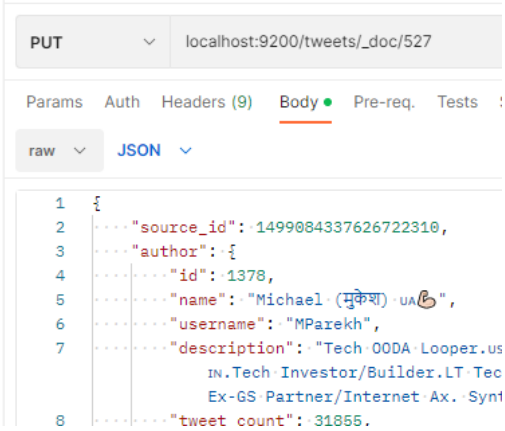
7. Experimentujte s nódami, a zistite koľko nódov musí bežať (a ktoré) aby vám Elasticsearch vedel pridávať dokumenty, mazať dokumenty, prezerať dokumenty a vyhľadávať nad nimi? Dá sa nastaviť Elastic tak, aby mu stačil jeden nód? Čo je dôvodom toho že existuje nejaké kvórum?

Táto úloha sa správala u mňa zvláštne. Teda čakal som, že čím viac nodov pobeží, tým rýchlejšie budú veci. Ale pravdou bolo, že keď bežal práve jeden, tak som mal najlepšie výsledky. Cvičiaci mi poradil, aby som si nastavil pre jeden shard aspoň 1 cpu. Táto zmena trocha zlepšila rozdiely. Avšak túto skutočnosť pripisujem tomu, že ak je viac nodov, treba omnoho viacej času na manažment vecí okolo a pri takto malej vzorke to prevýši čas potrebný na vyhľadávanie. Samozrejme, keď nebežal žiaden node, tak to nenašlo nič a dostal som error.
Pridávanie záznamu nevrátilo took ale uviedol som aspoň čas, ktorý sa zobrazil v postmanovi. Ak by sme chceli elastic len s jedným nodom, tak musí byť typu master aj data a je vhodné nastaviť na index len jeden shard.

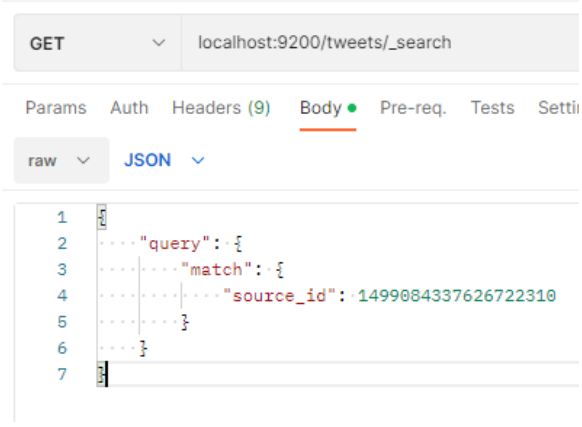
	Pridávanie	Mazanie	Prezeranie	Vyhľadávanie
Bežia 3 node-y	X (39ms)	37	5	7
Bežia 2 node-y	X (30ms)	34	6	10
Beží 1 nod-y (len master)	X (25ms)	21	5	7
Beží 0 nodov	X	X	X	X

Tabuľka trvaní (took)

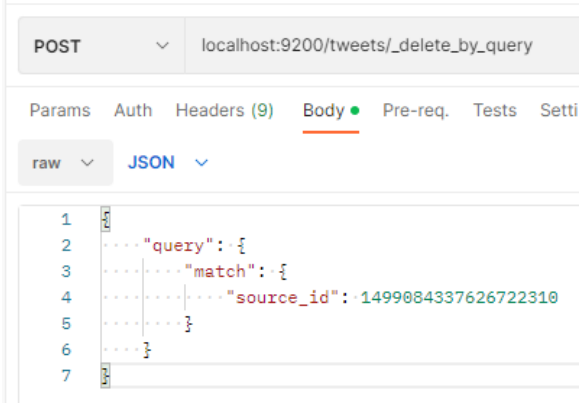
Pridávanie



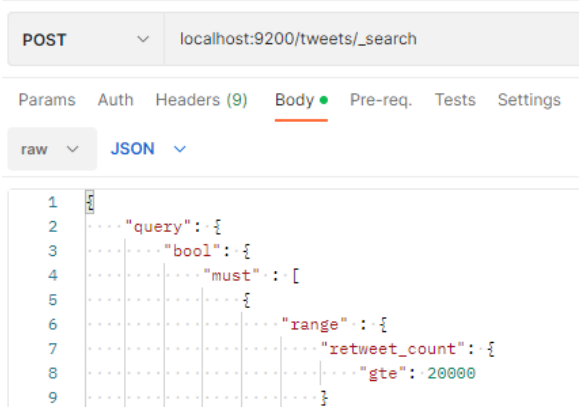
Prezeranie



Mazanie



Vyhľadávanie



Ukážka ako sa rozdelili shardy a repliky pri **dvoch** nodoch:

```
index  shard prirep state       docs store ip         node
tweets 0     p      STARTED    10925 2.8mb 172.19.0.2 es01
tweets 0     r      UNASSIGNED
tweets 1     p      STARTED    10969 2.8mb 172.19.0.4 es03
tweets 1     r      UNASSIGNED
tweets 2     r      STARTED    10818 2.8mb 172.19.0.2 es01
tweets 2     p      STARTED    10818 2.8mb 172.19.0.4 es03
```

```
ip         heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
172.19.0.2           23          98  12    0.53    0.81     0.75 dm        *      es01
172.19.0.4           39          95  12    0.53    0.81     0.75 d         -      es03
```

Ukážka ako sa rozdelili shardy a repliky pri **jednom** node:

```
index  shard prirep state       docs store ip         node
tweets 0     p      STARTED    10925 2.8mb 172.19.0.2 es01
tweets 0     r      UNASSIGNED
tweets 1     p      UNASSIGNED
tweets 1     r      UNASSIGNED
tweets 2     p      STARTED    10818 2.8mb 172.19.0.2 es01
tweets 2     r      UNASSIGNED
```

```
ip         heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
172.19.0.2           40          99   2    0.46    0.63     0.69 dm        *      es01
```

8. Upravujte počet retweetov pre vami vybraný tweet pomocou vašeho jednoduchého scriptu (v rámci Elasticsearchu) a sledujte ako sa mení _seq_no a _primary_term pri tom ako zabíjate a spúšťate nódy.

Vytvoril som skript na meneni retweet_count pre tweets s *„_id“* 1.

Vždy keď updatnem tento dokument, alebo aj ktorýkoľvek iný. Tak _seq_no sa zvýši o jedno. Je to preto, lebo toto číslo predstavuje počet operácií nad indexom. A teda keď som opätovne posielal rovnaký request, tak sa toto číslo zvyšovalo o jeden.

Čo sa týka _primary_term, tak tento sa nezvyšoval obyčajným posielaním reqestu. Pre jeho zmenu som skúšal vypnúť es-03 node. Ale nič sa nezmenilo a teda tento dokument nebol uložený na primary sharde na tomto node. Zmena nastala pri vypínaní es-02, es-01 a celého klastra. Vždy pri týchto procesoch sa záznam zapísal na iný shard a vtedy sa zmenil aj _primary_term.

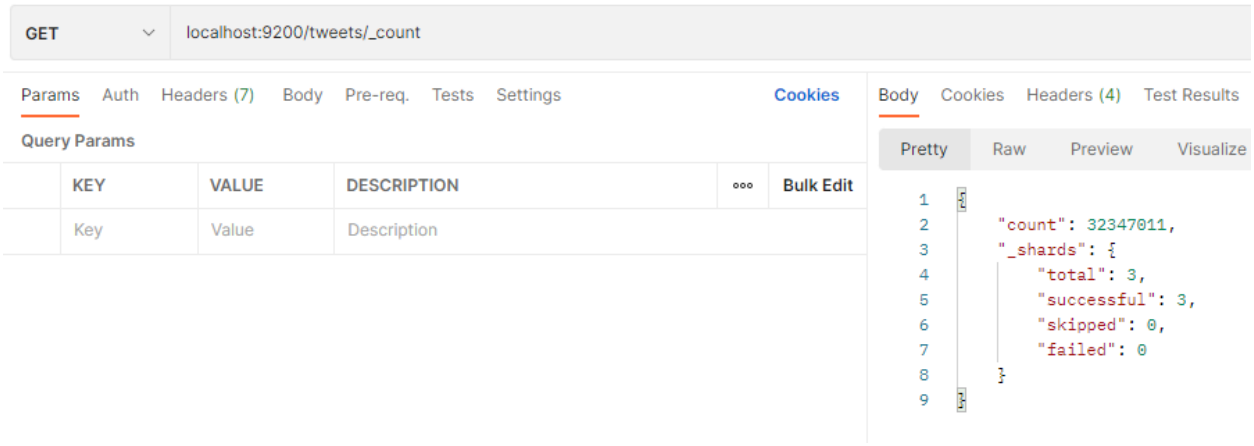
Tieto dve čísla elastic používa na to aby zabezpečil Optimistic concurrency control. V skratke ide o riešenie konfliktov pri lost update problému.

Pred vypnutím es-02

Po vypínaní es-02, es-01 a celého klastra

9. Zrušte repliky a importujete všetky tweety

Vytvoril som si nový index bez replík a starý som vymazal. Pomocou rovnakého java skriptu ako v úlohe 5 som si naimportoval všetky dáta, ktoré som mal exportnuté do jsonu. Import trval cca 6 hodín.

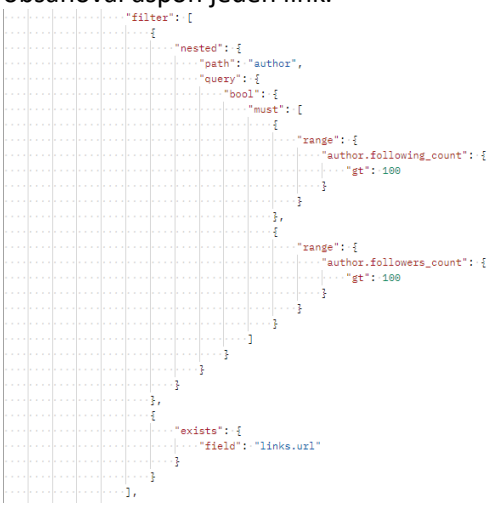


10. Vyhľadajte vo vašich tweetoch, kde použite function_score pre jednotlivé medzikroky nasledovne

V query na vyhľadanie „put1n chr1stian fake jew“ som použil multi_match, kedže trebalo viacero fieldov. Type som nastavil na most_fields, aby sa dokumenty zoraďovali podľa početnosti výrazov. Boost pre jednotlivé fielde som spravil cez ^. Ďalej do must som pridal term pre hashtag a pre conversation_references.content.

```
{
  "size": 10,
  "query": {
    "function_score": {
      "query": {
        "bool": {
          "must": [
            {
              "multi_match": {
                "query": "put1n chr1stian fake jew",
                "type": "most_fields",
                "fields": [
                  "author.description.shingles^10",
                  "content^6"
                ],
                "fuzziness": "auto",
                "operator": "or"
              }
            },
            {
              "term": {
                "hashtags": "ukraine"
              }
            },
            {
              "term": {
                "conversation_references.content": "nazi"
              }
            }
          ],
```

Čo sa týka filtrov, tak som potreboval urobiť nested query pre autora, kde som naraz vyfiltroval dané počty pre following_count a followers_count. Ako posledný filter som ešte pridal to, aby dokument obsahoval aspoň jeden link.



```
          "filter": [
            {
              "nested": {
                "path": "author",
                "query": {
                  "bool": {
                    "must": [
                      {
                        "range": {
                          "author.following_count": {
                            "gt": 100
                          }
                        }
                      },
                      {
                        "range": {
                          "author.followers_count": {
                            "gt": 100
                          }
                        }
                      }
                    ]
                  }
                }
              }
            },
            {
              "exists": {
                "field": "links.url"
              }
            }
          ],
```

Do časti should som pridal 4 pravidlá. Dva nested query pre contex_domain a context_entity, ktoré som boostol podľa zadania.

```
"should": [
    {
        "nested": {
            "path": "context_annotations",
            "query": {
                "nested": {
                    "path": "context_annotations.entity",
                    "query": {
                        "match": {
                            "context_annotations.entity.name": {
                                "query": "Soros",
                                "boost": 10
                            }
                        }
                    }
                }
            }
        }
    },
    {
        "nested": {
            "path": "context_annotations",
            "query": {
                "nested": {
                    "path": "context_annotations.domain",
                    "query": {
                        "match": {
                            "context_annotations.domain.name": {
                                "query": "Person",
                                "boost": 5
                            }
                        }
                    }
```

Ďalej nested query pre autora a pre content , kde som boostol daný dokument pokiaľ obsahoval frázu „put1n chr1stian fake jew“ so slopom 1 (výmena slov).

```
                {
                    "nested": {
                        "path": "author",
                        "query": {
                            "match_phrase": {
                                "author.description": {
                                    "query": "put1n chr1stian fake jew",
                                    "slop": 1,
                                    "boost": 5
                                }
                            }
                        }
                    }
                },
                {
                    "match_phrase": {
                        "content": {
                            "query": "put1n chr1stian fake jew",
                            "slop": 1,
                            "boost": 5
                        }
```

Nakoniec som pridal agregáciu, kde som agregoval dokumenty podľa týždňov. Spočítal som celkový počet dokument, ktoré spadajú do tohto týždňa a potom som vytvoril bucket pro-russia, ktorý zrátal, koľko z nich obsahovalo hashtagy zo zadania.

```
"aggs": {
    "hashtags": {
        "date_histogram": {
            "field": "created_at",
            "calendar_interval": "week",
            "format": "yyyy-MM-dd"
        },
        "aggs": {
            "hashtags": {
                "filters": {
                    "filters": {
                        "pro-russia": {
                            "terms": {
                                "hashtags": [
                                    "istandwithputin",
                                    "racism",
                                    "1trillion",
                                    "istandwithrussia",
                                    "isupportrussia",
                                    "blacklivesmatter",
                                    "racistukraine",
                                    "africansinukraine",
                                    "palestine",
                                    "israel",
                                    "freepalestine",
                                    "istandwithpalestine",
                                    "racisteu",
                                    "putin"
                                ]
```

Nakoniec som vyfiltroval také buckety, ktoré neobsahovali ani jeden dokument, kvôli lepšej prehľadnosti vo výsledku.

```
            }
        },
        "sales_bucket_filter": {
            "bucket_selector": {
                "buckets_path": {
                    "docCount": "_count"
                },
                "script": "params.docCount > 0"
            }
        }
```

Daná query mi našla tento dokument, ktorý má najväčšie skóre:


```
"hits": {
    "total": {
        "value": 1371,
        "relation": "eq"
    },
    "max_score": 142.47322,
    "hits": [
        {
            "_shard": "[tweets][1]",
            "_node": "S-BeD_JVRSCqzpuec0moyQ",
            "_index": "tweets",
            "_id": "10477411",
            "_score": 142.47322,
            "_source": {
                "source_id": 1497698752987390979,
                "author": {
                    "id": 1450394162,
                    "name": "us Reel Conquest us",
                    "username": "ReelConquest",
                    "description": "us US Army Veteran us Patriot, 2nd
                        Amendment advocate, MAGA #TrumpWon🦅🌳🇺🇸
                        us🇺🇸🚂🚒🚓🚗",
                    "tweet_count": 36466,
                    "listed_count": 3,
                    "followers_count": 906,
                    "following_count": 1115
                },
```


A agregácie vyzerali takto:


```
{
    "key_as_string": "2022-03-07",
    "key": 1646611200000,
    "doc_count": 86,
    "hashtags": {
        "buckets": {
            "pro-russia": {
                "doc_count": 48
            }
        }
    }
},
{
    "key_as_string": "2022-03-14",
    "key": 1647216000000,
    "doc_count": 206,
    "hashtags": {
        "buckets": {
            "pro-russia": {
                "doc_count": 104
            }
        }
    }
},
{
    "key_as_string": "2022-03-21",
    "key": 1647820800000,
    "doc_count": 201,
    "hashtags": {
        "buckets": {
            "pro-russia": {
                "doc_count": 112
            }
        }
```