Elastic Search

"wat heb je aan data als je er niets mee doet.."

Oscar Buse

11 juli 2017

Linux User Group Nijmegen

Inleiding

Dit praatje gaat over Elasticsearch.
De onderwerpen die aan bod komen:

- Wat is Elasticsearch?
- Waarom Elasticsearch?
- Installatie.
- Terminologie.
- Wat is je data en hoe wordt deze data opgeslagen?
- Werken met Elasticsearch. Theorie met voorbeelden
- Een toepassing.
- De voor- en nadelen van Elasticsearch.
- Referenties.

Wat is Elasticsearch?

Definitie: gedistribueerde tekst zoek (en analyse) software met http interface ("RESTful") en json format.
Enkele eigenschappen:

- Zeer snel (**alles** wordt default ge-indexeerd).
- Geheugen intensief.
- ES is document (object) georienteerd (ipv rij/kolom georienteerd).
- Goed voor full text search.
- Ingebouwde score bij relevantie.
- Ingebouwde *highlighting*.
- Distributed (horizontaal schaalbaar, meerdere nodes).
- Ook voor data aggregatie en analyse.

Is allemaal niet nieuw maar het zit wel allemaal samen in 1 applicatie.

Waarom Elasticsearch?

Zaak om eerst te kijken wat je kunt hebben aan ES.

- Zie o.a. de eerder genoemde eigenschappen.
- Je hebt niks aan data als je er niet wat mee doet.
- Leegte in log analyse.. Splunk kan bv. veel maar is duur..
- Open Source.
- Maken van *data-informed* beslissingen.

Wat kan ES wat bv. niet kan in SQL?
Niet de juiste vraag: gebruik de tool die voldoet voor jouw situatie.
Eenmaal wat meer bekend geef ik een opsomming van de voor- en nadelen.

Installatie

- Eenvoudig met downloaden van tgz-file of packages (rpm of deb).
- Default luistert Elasticsearch op 127.0.0.1, port 9200.
- Ook bij een cluster: 1 interface naar je data.
- Config in /etc/elasticsearch
- OS configuratie, o.a.:
  - disable swap (gebruik *ES only* servers/VM's). Only 1st slide
  - zorg voor genoeg open files (max aantal file descriptors, /proc/security/limits.conf).
  - zorg voor genoeg mogelijke processen (ulimit -u).

  Veel wordt al gedaan via de package installers. Check settings met de documentatie (zie referenties).
- verschil development en productie omgeving (bv. met setting network.host)

Terminologie

Lucene	ES is afgeleid van Apache's Lucene: software voor text indexering en zoeken.
cluster	een verzameling van servers (nodes) wat elasticsearch mogelijk maakt.
node	1 server (of VM) als onderdeel van een cluster.
index	verzameling data met overeenkomstige kenmerken. Vergelijk met een **database** in SQL.
type	verzameling data binnen een index. Vergelijk met een **table** in SQL.
field	veld binnen een type. Vergelijk met **kolom** in SQL.
document	de basis-eenheid van informatie in ES. Geformatteerd in JavaScript Object Notatie (JSON).

Relational DB => Databases => Tables => Rows => Columns

Elasticsearch => Indices => Types => Documents => Fields

term een unieke ge-indexeerde waarde ("Bla" is wat anders dan "bla")

shard

- unit welke een deel van alle data bevat (goed voor opdelen en verdelen van data).
- 1 shard kan maximaal 2.147.483.519 *documenten* bevatten.
- primary en replica shards.
- aantal primary shards bepaald hoe groot je db kan zijn.
- replica shards zijn readonly. Goed voor redundancy en performance (parallel gebruik van replicas).
- default 5 primary en 5 replica shards per index (db).
- minimaal 1 shard per node.
- aantal primary shards ligt vast, aantal replicas kan gewijzigd worden.

Data opslag

Hoe wordt de data opgeslagen?

- opslag op het filesysteem.
- veel gebruik gemaakt van compressie.
- data in JSON records.
- mapping voor type velden (default: text + keyword (vroeger: string)).
- **Alles** (elk "field") wordt ge-indexeerd en is doorzoekbaar.
- indexeren met "inverted" indexen: ("terms" naar documents).
- verschil in "full text fields" (match search) en "exact value fields" (term search).
  - *analyzed* en *not analyzed*
  - Bv. string => text (*analyzed*) + keyword (exact, *not analyzed*)

Werken met ES

Data wordt opgeslagen, opgevraagd, gewijzigd met een REST API (HTTP requests), denk hierbij aan bv.:

- Document API: voor CRUD operaties.
- Cluster API: checken van health, status en verkrijgen van statistieken van je cluster/nodes
- Index API: beheer je database (bv. DELETE, CREATE).
- Search API: doe (geavanceerde) zoek acties.

Diverse hulpmiddelen voor deze API's: curl, python/perl modules, websites (Kibana), ...

Document API

Deze API ondersteunt diverse CRUD operaties.

- Url heeft de vorm:

```
POST|PUT|GET|DELETE host:9200/index/type/id -d { }
```

- POST: index (insert met auto-created id)

```
curl -XPOST '127.0.0.1:9200/lugn/mensen'
  -d '{"naam":  "Harry Nak"}'
```

- PUT: index (insert, geef zelf id op), update (index met zelfde id)

```
curl -XPUT '127.0.0.1:9200/lugn/mensen/1'
  -d '{"naam":  "Harry Nak"}'
```

- GET: Bv. een hele index (database) doorzoeken kan met:

```
curl -XGET 127.0.0.1:9200/lugn/_search?q=zoekstring
```

- DELETE

```
curl -XDELETE 127.0.0.1:9200/lugn?pretty'

curl -XDELETE 127.0.0.1:9200/lugn/mensen/2?pretty'

curl -XPOST
  127.0.0.1:9200/lugn/mensen/_delete_by_query?pretty \
  -d '{"query":{"match":{"naam":"Harry"}}}'
```

- UPDATE: (onder de motorkap DELETE en CREATE)

```
curl -XPOST 127.0.0.1:9200/lugn/mensen/1/_update \
  -d '{"doc":{"naam":"Harry A. Nak"}}'
```

Zoektechnieken: query-string search en "(json) request body" search.

- woord, woo* woor?
- fieldname:woord (bv. usernaam:harry)
- query (match + scores) en filters (match yes or no, geen scores)
- match (analyzed, text) en term (exact, keyword)) queries.
- aggregate, average, sort
- meer .. (Query DSL: flexibele query language in json format. Bv.:

```
"query":{
  "filter":{
    "geo_distance":{
      "distance":"100km",
      "location":[32.052098, 76.649294]
    }
  }
}
```

- meer .. (boosts, suggestions, completion)

Een "toepassing"

- sql2es.py
- search.py

De voor- en nadelen van Elasticsearch.

- - veiligheid? (geen authenticatie).
- - geen transactie (commits/rollback) model: minder controle over consistentie van de data.
- - relatief nieuw ("moet zich nog bewijzen").
- - query DSL kan erg complex worden (maar geldt bv. ook voor SQL).
- + Zeer schaalbaar, snel (alles is een index).
- + Lage drempel maar ook zeer flexibele en geavanceerde mogelijkheden.
- + veel OOTB: full text search, scoring, aggregatie, suggestie, "result highlighting".

De voor- en nadelen van Elasticsearch.

- + goede integratie met Logstash en Kibana (ELK-stack).
- + object storage (ipv key/value).
- + veel geolocatie "features".

Als met alles gebruik de juiste tool voor de juiste toepassing, bv.:

- betalingstransacties in een RDBMS
- zoeken van text in veel documenten: Elasticsearch

Voorbeelden van toepassingen:

- search engines voor bv. websites.
- search en analyseren van allerlei logging (weblog, syslog, klant-data, ...).
- wikipedia, the guardian, stack overflow, github, reisavonturen, ...

Referenties

- Website: `https://www.elastic.co`
- Leren: `https://www.elastic.co/learn`
- Setup/config: `https://www.elastic.co/guide/en/elasticsearch/reference/current/setup.html`
- OS settings: `https://www.elastic.co/guide/en/elasticsearch/reference/current/system-config.html`
- Cheatsheet: `http://elasticsearch-cheatsheet.jolicode.com/`

Bier?