Elastic ELK

Elasticsearch, Logstash & Kibana

Search & Analyze log files in clusters & Real Time

Berney Léonard, Bron Sacha, Minder Valentin, Salathe Fabien
HEIG-VD - CLD - 2016

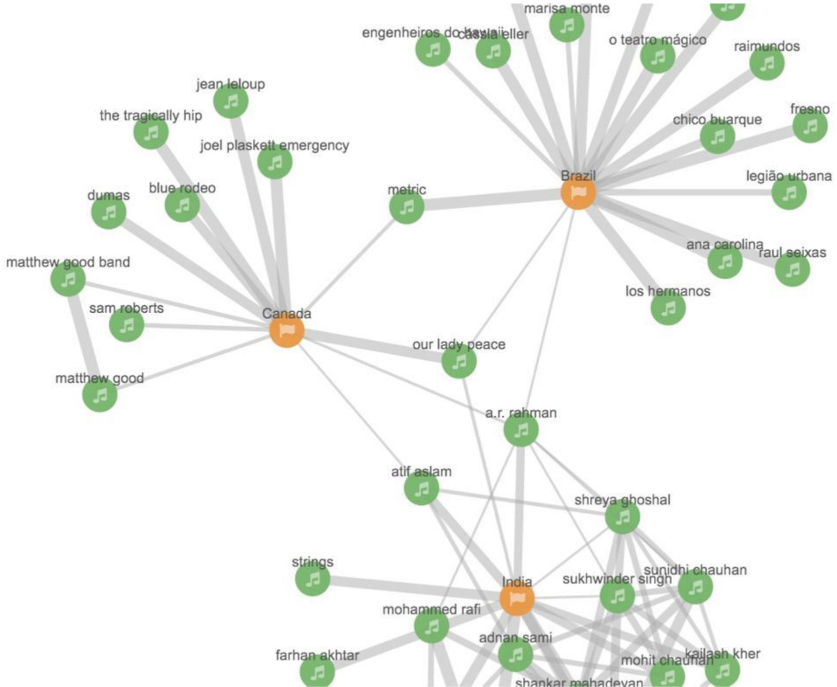
Table of contents

https://github.com/BafS/ElasticSearchExperiment

Introduction

Elasticsearch is a **search server**.

Elasticsearch is **distributed** and frequently used via a **web page.**



Stack "ELK"

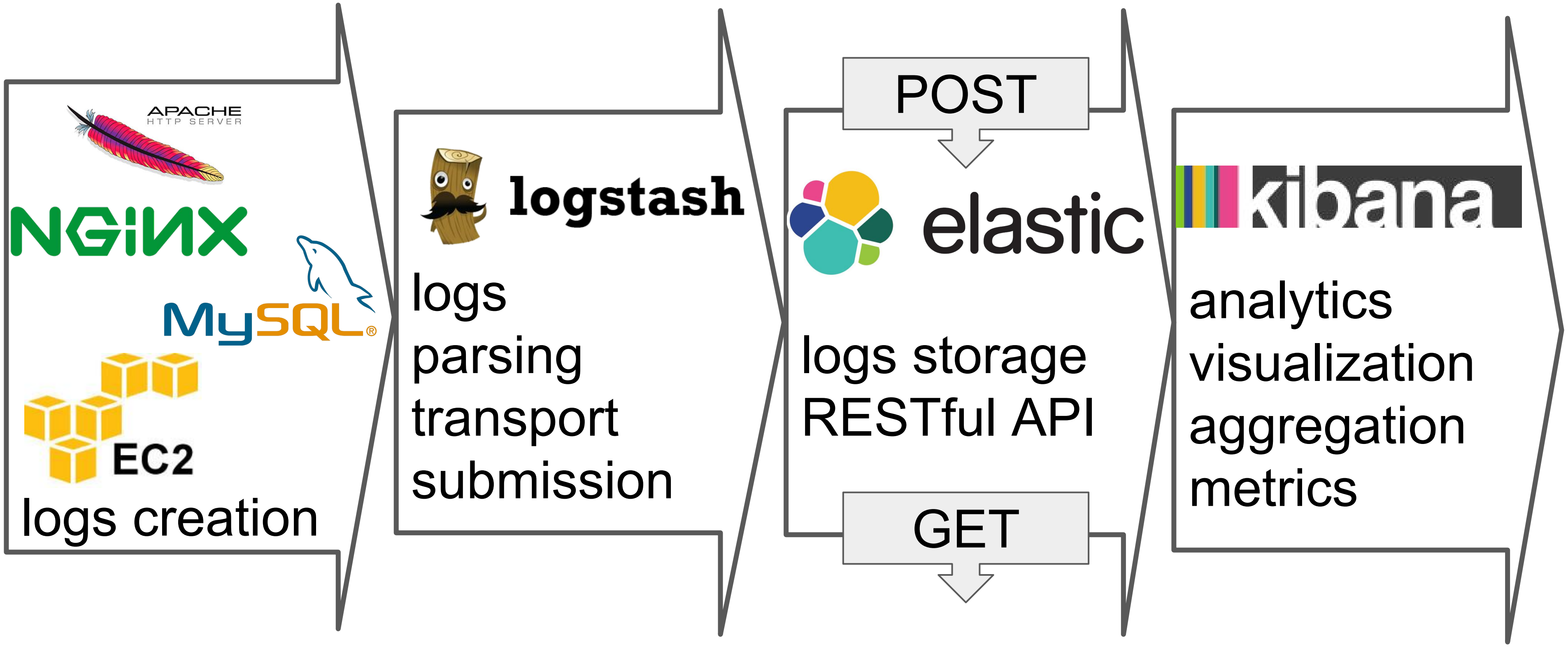
- Your existing servers
- Logstash
- Elasticsearch
- Kibana

Stack "ELK"

- Logstash: abstraction and wrappers with for **logs parsing**
- Elasticsearch: **RESTful API** for log storage & retrieval
- Kibana: real-time data gathering, **visualization** and aggregation

elastic logstash kibana

Stack "ELK" in context



logstash

Logstash is a tool used to collect, handle and the push events and log messages to the server.

- Logs input can be customised to "watch" any log file.
- Filters can be created to handle any type of log structure.
- Processed information are then push to the server, Elasticsearch.

logstash

```
# logstash.conf
input {
  file {
    path => "/var/log/apache2/access_log"
    type => "apache_access_log"
  }
  file {...}
}

filter {
  if [type] == "apache_access_log" {
    mutate { replace => { "type" => "apache-access" } }
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
    date {
      match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]
    }
  }
  if [type] == "apache_error_log" {...}
  if [type] == "syslog" {...}
  }
}

output {
  elasticsearch { hosts => "search-elastic-search-heig-superman.amazonaws.com:80" }
}
```

```
patterns.d/apache-error
APACHE_ERROR_LOG \[(?<timestamp>%{DAY:day} %{MONTH:month} %{MONTHDAY}
%{TIME} %{YEAR})\] \[.*:%{LOGLEVEL:loglevel}\] \[pid %{NUMBER:pid}\] \
[client %{IP:clientip}:.*\] %{GREEDYDATA:message}
```

elastic

How it works ?

- Ready and easy-to-use and simple RESTful JSON API
- Stores everythings
- Retrieves everything, selected fields, or aggregations (sum, max, avg. etc)
- Accepts any input within the PUT (fields don’t need to be declared)
- PUT on server.com/index/ressource/ creates a NEW ressource under index pattern (a single server may have several indexes)
- By default, no security, no authentication (plugin/add-on available)
- OK for log display, analysis & visualisation, not sensitive data.

elastic

Advantages

- RESTful JSON API
- Fast deployment
- Aggregation
- Designed to be distributed
- Open source
- Query DSL syntax is really flexible and easy to use: general search or specific field / choose what fields to output / usual condition (AND, OR, NOT, ranges)
- even easier with kibana visualisation
- Official Javascript npm package

elastic

Drawbacks

- Still relatively **new project**, not a lot of documentations, maybe not very stable.
- **Security**: ElasticSearch does not provide any authentication or access control functionality.
- **Transactions**: There is no support for transactions or processing on data manipulation
- **Durability**: ES is distributed and fairly stable but backups and durability are not as high priority as in other data stores. This is probably the most important if you're going to make ES the primary store since losing your data is never good.

elastic

```
curl -XPUT 'https://example.com/logs/windows/bug12421?pretty' -d '{ "err" :
"Path too long" }'

curl -XGET 'https://example.com/logs/windows/_search?q=Path&pretty=true'
{
  "took" : 3, "timed_out" : false, "_shards" : { "total" : 5, "successful" : 5, “failed" : 0 },
  "hits" : {
    "total" : 1, “max_score" : 0.15342641, "hits" : [{
      "_index" : "logs",
      "_type" : "windows",
      "_id" : "bug12421",
      "_score" : 0.15342641,
      "_source": { "err" : "Path too long" }
    }]
  }
}
```

elastic

Demo 1: How to Use Elasticsearch

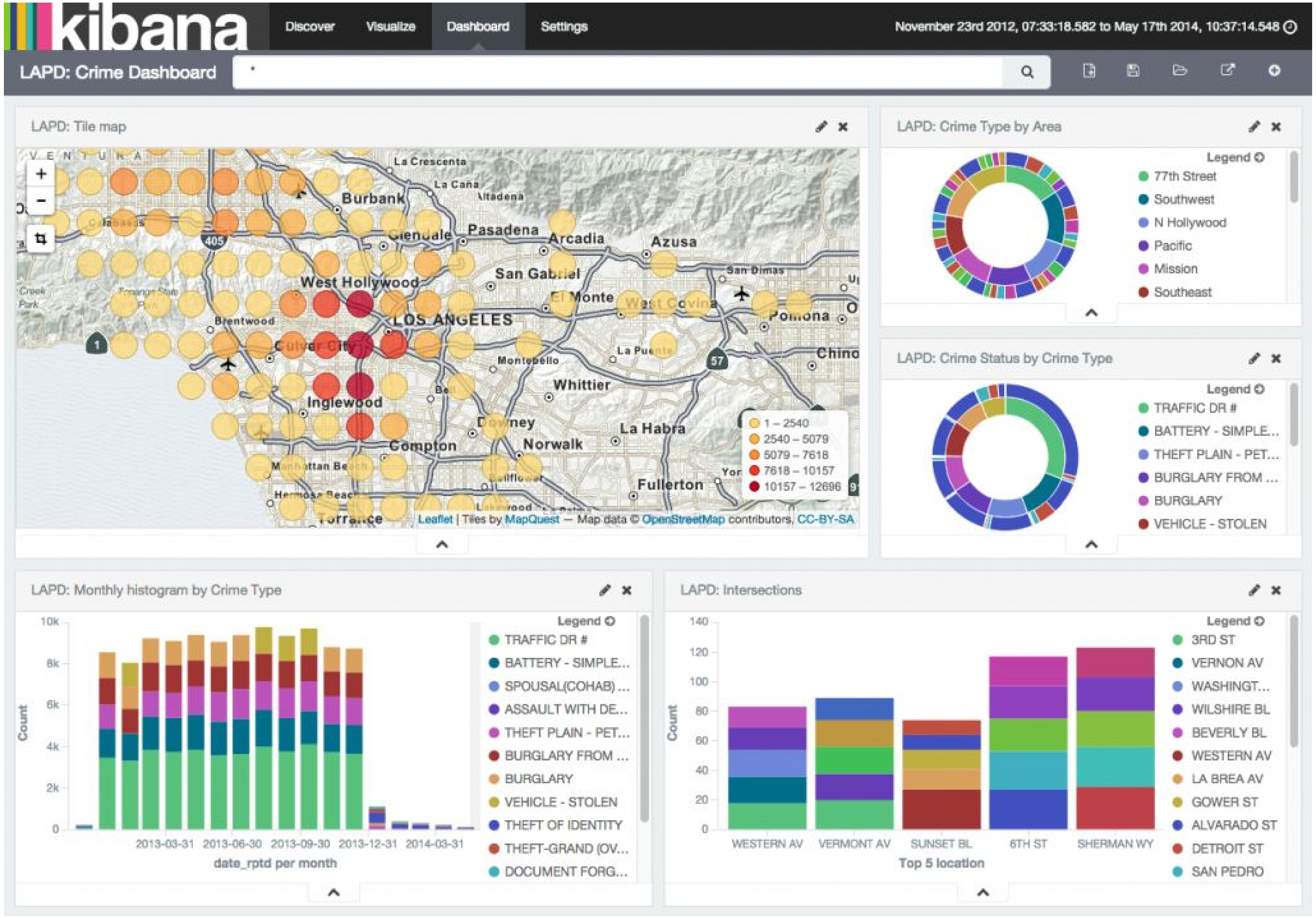
- Servers (Apache, database, system ...) send their data to the ELS
- Create some errors/logs (db shutdown, wrong login, kill processes etc)
- Explore the logs to find the issues
- The same ELS contains logs from multiple sources
- The logs update automatically & regularly

https://search-elastic-search-heig-3nhbodzwhflo56pew23jotan6a.eu-central-1.es.amazonaws.com/_plugin/kibana/#

kibana

Aggregation & Visualisation Software

- Discover : search & aggregate
  - search anywhere
  - search by field, toggle arguments
  - view the field you choose
- Visualisation : charts
  - pie chart, area chart, line chart, vertical chart, metrics, tile map etc.
  - from any (cross-) search possible
- Dashboard : home screen for saved visualisations



Los Angeles Police Department Crime Dashboard (source: www.elastic.co)

Demo 2: Aggregation & Visualisation of Public Transportation Stop

- raw set from Opendata
- little ameliorations (formats of dates, numbers, GPS coordinates)
- Intersting fields
  - Stop name (Yverdon*): `Nom:Yverdon*`
  - Municipality (Commune): `NomCommune:Lausanne`
  - Company (CFF, CarPostal, TL, TPG…): `AbreviationET:sbbcffffs`
  - Type (bus, cheminfer, tram, telepherique, bateau, funiculaire, cremaillere, metro): `MoyenTransport`
  - Type d'exploitation (Arret/PointChargement): `TypePointExploitation`
  - Altitude above see level: `Altitude:[1000 TO *], Altitude:>=500`
  - GPS Coordinate: `location`

https://opendata.swiss/en/dataset/haltestellen-des-offentlichen-verkehrs

xtf_id	ch14uvag00064399	#unique ID
Numero	8504200	#Stop
Nom	Yverdon-les-Bains	Full stop name
Abreviation	YV	Abbreviation
RespDonneesAbreviation	SBBCFFFFS	Entreprise: + empty
NumeroET	1	#Entreprise: [1-9999]
AbreviationET	SBBCFFFFS	Entreprise: pag (CarPostal), sbbcffffs, vbz, tpg, tpf, tl, vzo, trn, bvb, zvb, bos, vbg, svb, blt, rbs, bls, sti, fr, rhb, vr, etc...
TypePointExploitation	Arret_et_PointChargement	Type: Arret / PointChargement / PointExploitationSimple
MoyenTransport	CheminFer	Type: bus, cheminfer, tram, telepherique, bateau, funiculaire, cheminfercremaillere, metro
Altitude	435	Meters ASL: [186-3454]
NumeroCommune	5938	OFS Number [1-9999]
NomCommune	Yverdon_les_Bains	Municipality
y_Coord_Est, x_Coord_Nord	539100, 181500	CH1903/LV03 - Swiss Army Coordinates
location	{"lat":46.7818,"lon":6.6411}	WGS84 GPS Coordinate (computed from CH1903*)

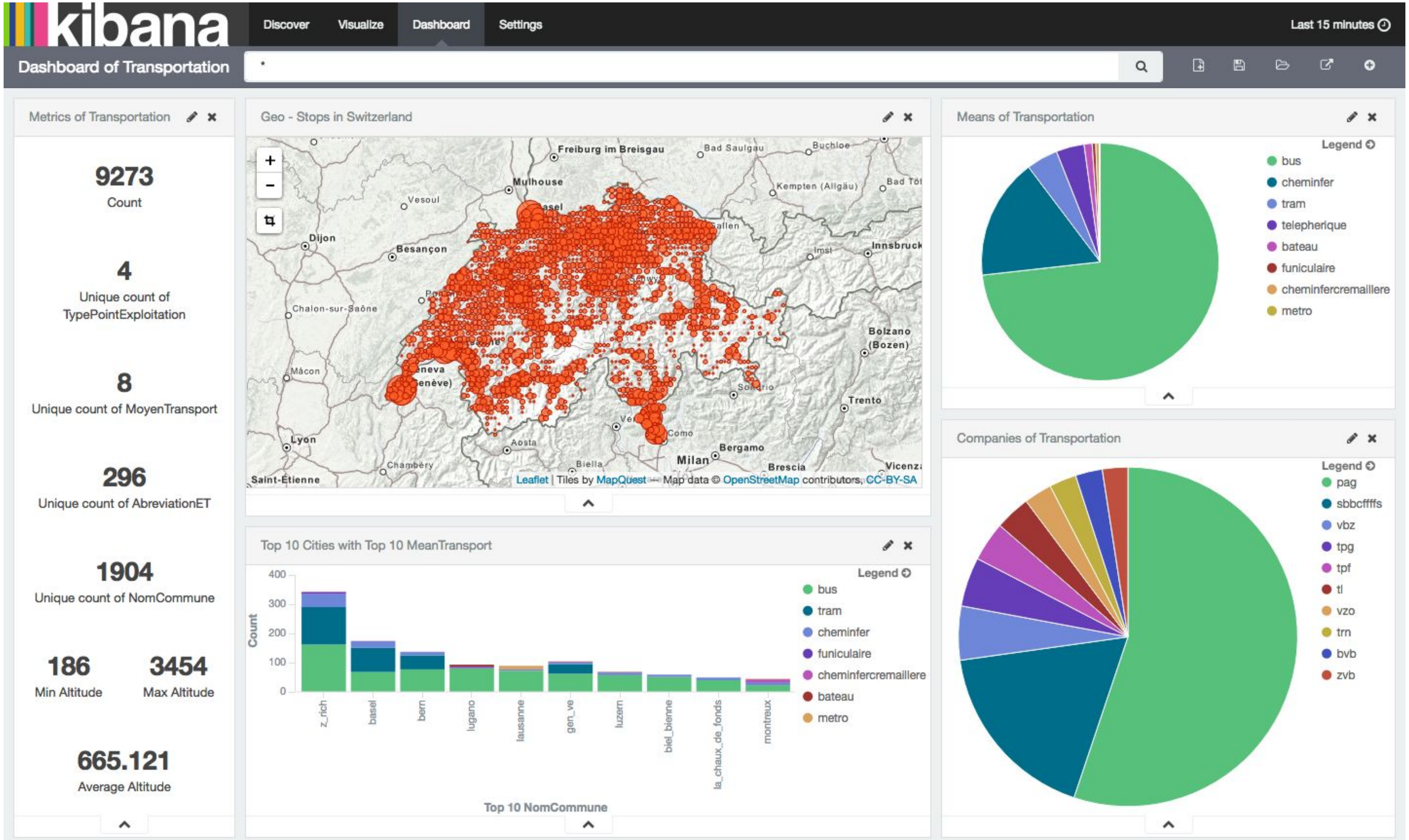
Other date info, not shown: DebutValidite, FinValidite, DateTraitement, Etat
* http://www.swisstopo.admin.ch/internet/swisstopo/fr/home/products/software/products/skripts.html

Demo 2: Aggregation & Visualisation of Swiss Public Transportation Stops

- dashboard, basics metrics: number of unique … by “field”
- pie charts, bar charts: count number of … by “field”
- GPS visualisations
- cross-requests: Top 5 X by Y/ for Z
  - Top 10 Means by Top 10 Cities
  - Wrong Visualisation may hide data!
- Togglable fields & Make your own query !
  - Differences between `NomCommune:"Lausanne"` and `Nom:"Lausanne"` (analyzed fields)
  - `NomCommune:"Lausanne" AND/OR NOT MoyenTransport: Metro`
  - `_exists_:"var"  / _missing_:"var"`
  - `+ term - term`

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html?q=query%20string#_field_names

Kibana - Dashboard - Swiss Public Transportation Stops Visualisations



Metrics of Transportation

9273
Count

4
Unique count of TypePointExploitation

8
Unique count of MoyenTransport

296
Unique count of AbreviationET

1904
Unique count of NomCommune

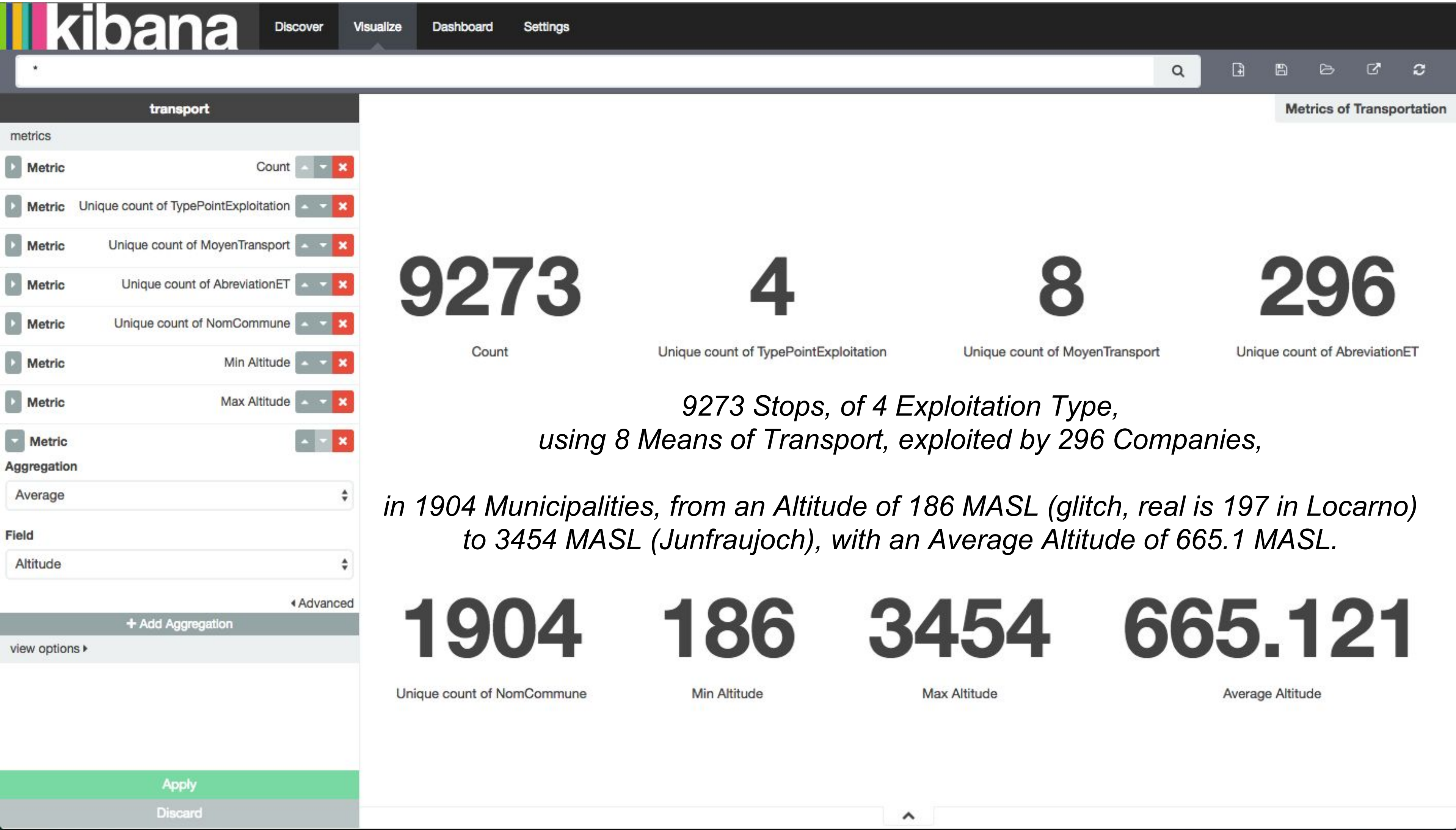
186
Min Altitude

3454
Max Altitude

665.121
Average Altitude

https://search-elastic-search-heig-3nhbodzwhflo56pew23jotan6a.eu-central-1.es.amazonaws.com/_plugin/kibana/#/dashboard/Dashboard-of-Transportation

Kibana - Global Metrics



Metric	Count
Metric	Unique count of TypePointExploitation
Metric	Unique count of MoyenTransport
Metric	Unique count of AbreviationET
Metric	Unique count of NomCommune
Metric	Min Altitude
Metric	Max Altitude
Metric	

Aggregation: Average

Field: Altitude

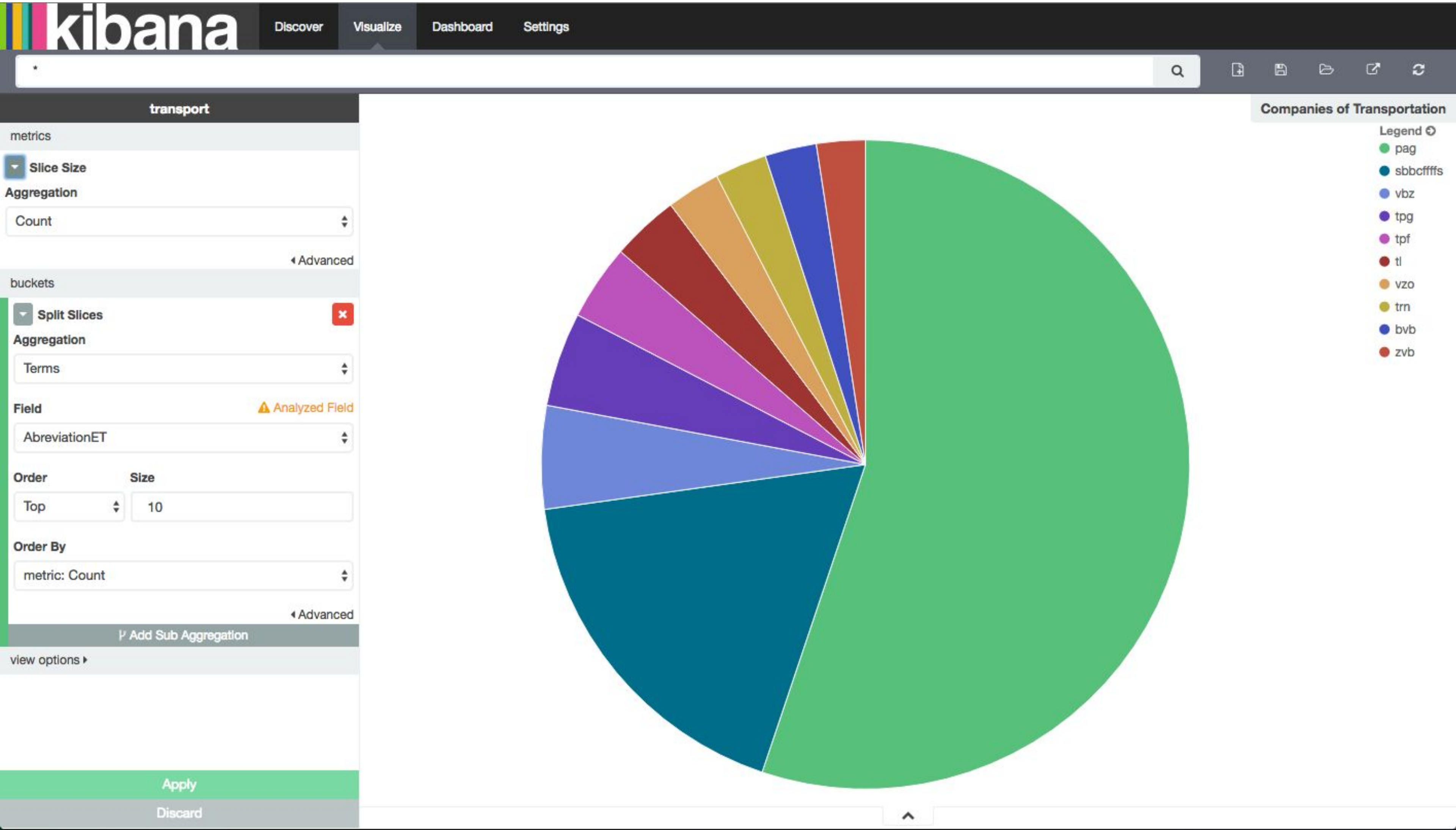
Count	Unique count of TypePointExploitation	Unique count of MoyenTransport	Unique count of AbreviationET
9273	4	8	296

9273 Stops, of 4 Exploitation Type,
using 8 Means of Transport, exploited by 296 Companies,

in 1904 Municipalities, from an Altitude of 186 MASL (glitch, real is 197 in Locarno)
to 3454 MASL (Junfraujoch), with an Average Altitude of 665.1 MASL.

Unique count of NomCommune	Min Altitude	Max Altitude	Average Altitude
1904	186	3454	665.121

Kibana - Pie Chart - Companies of Transportation



Slice Size

Aggregation: Count

Split Slices

Aggregation: Terms

Field: AbreviationET ⚠ Analyzed Field

Order: Top

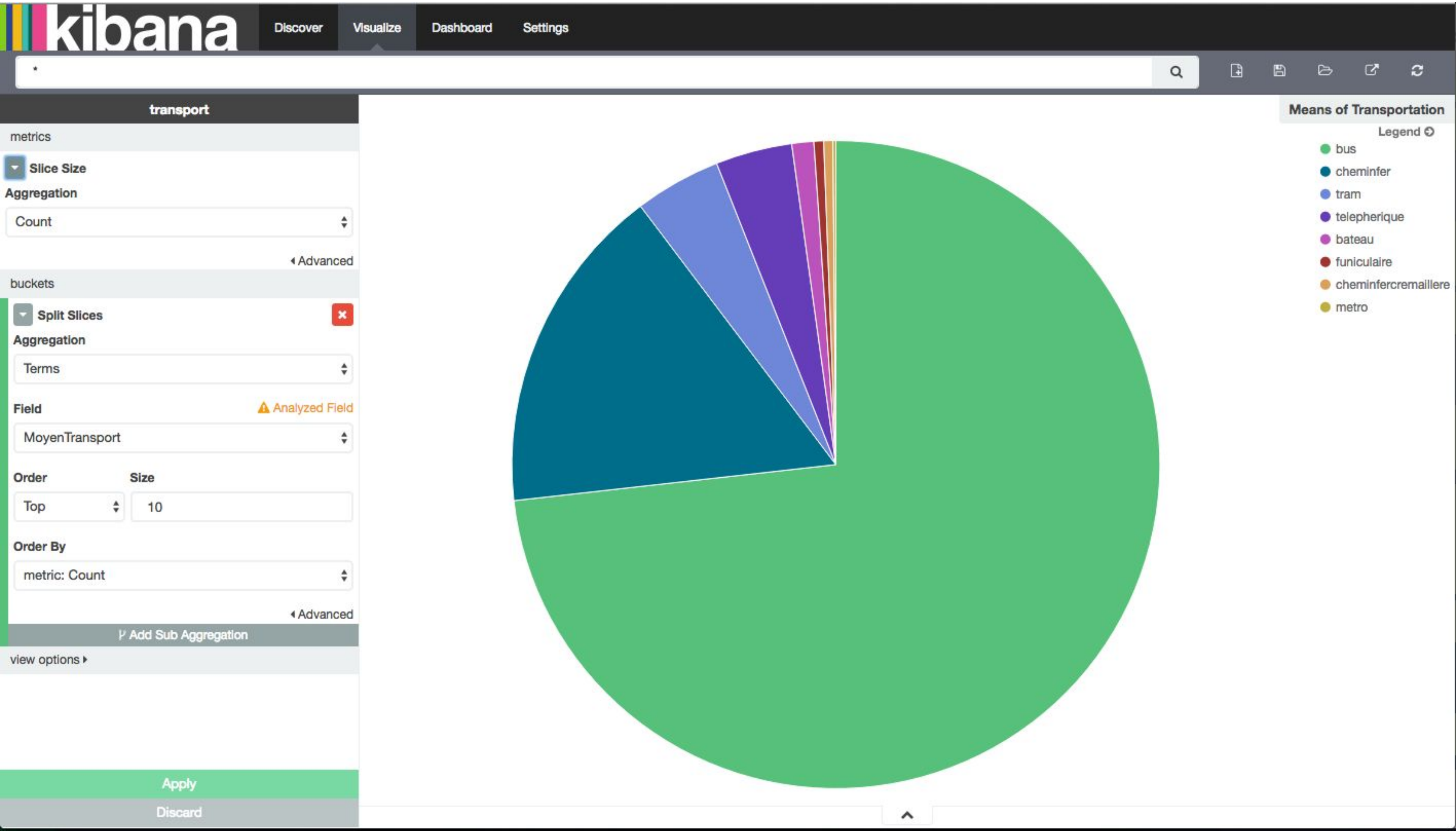
Size: 10

Order By: metric: Count

Companies of Transportation

Legend
- ● pag
- ● sbbcffffs
- ● vbz
- ● tpg
- ● tpf
- ● tl
- ● vzo
- ● trn
- ● bvb
- ● zvb

Kibana - Pie Chart - Means of Transportation



Aggregation: Count

Aggregation: Terms

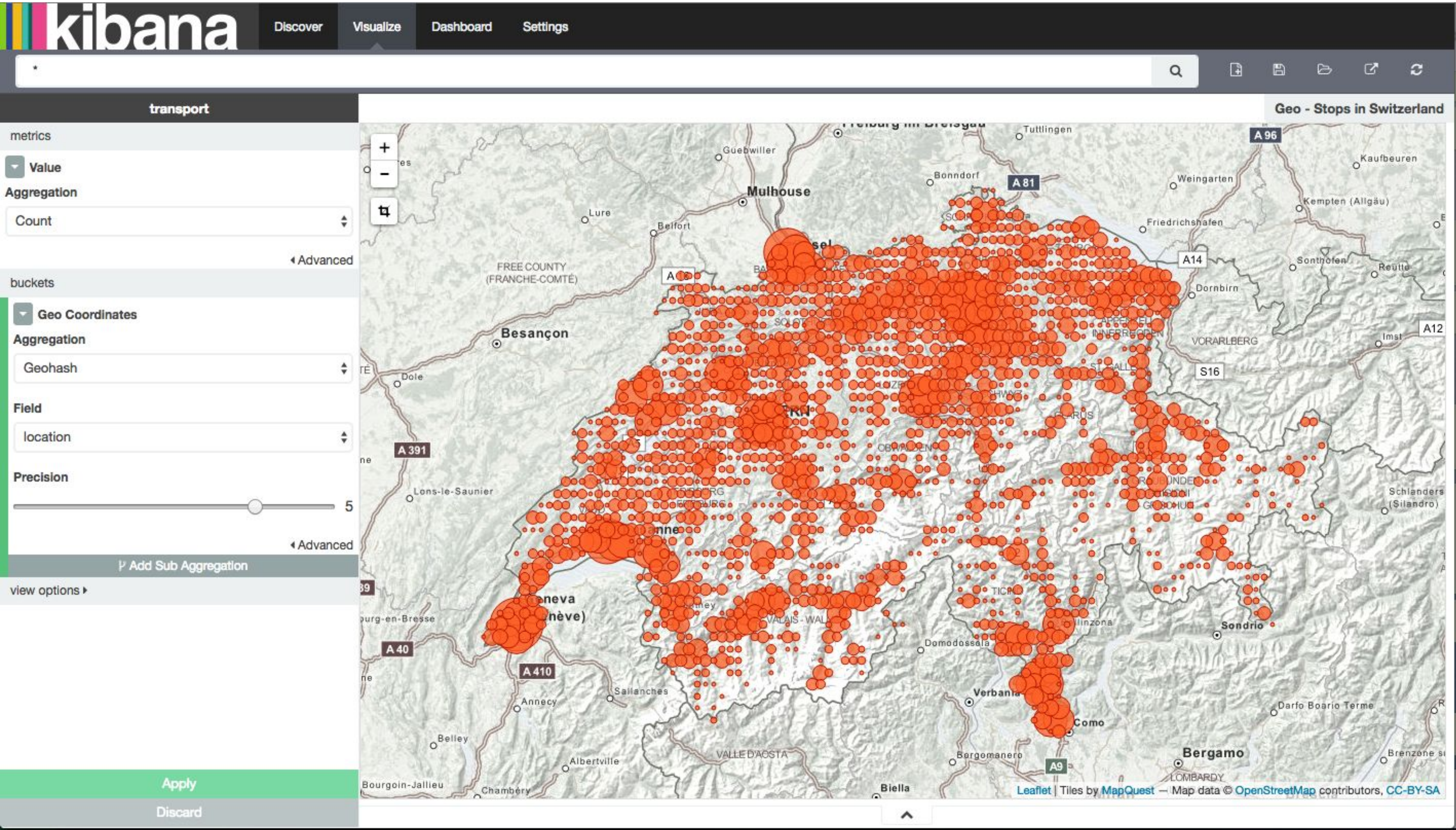
Field: MoyenTransport

Order: Top

Size: 10

Order By: metric: Count

Kibana - Tile Map (Geodata)



Discover Visualize Dashboard Settings

transport

Geo - Stops in Switzerland

metrics

Value

Aggregation: Count

buckets

Geo Coordinates

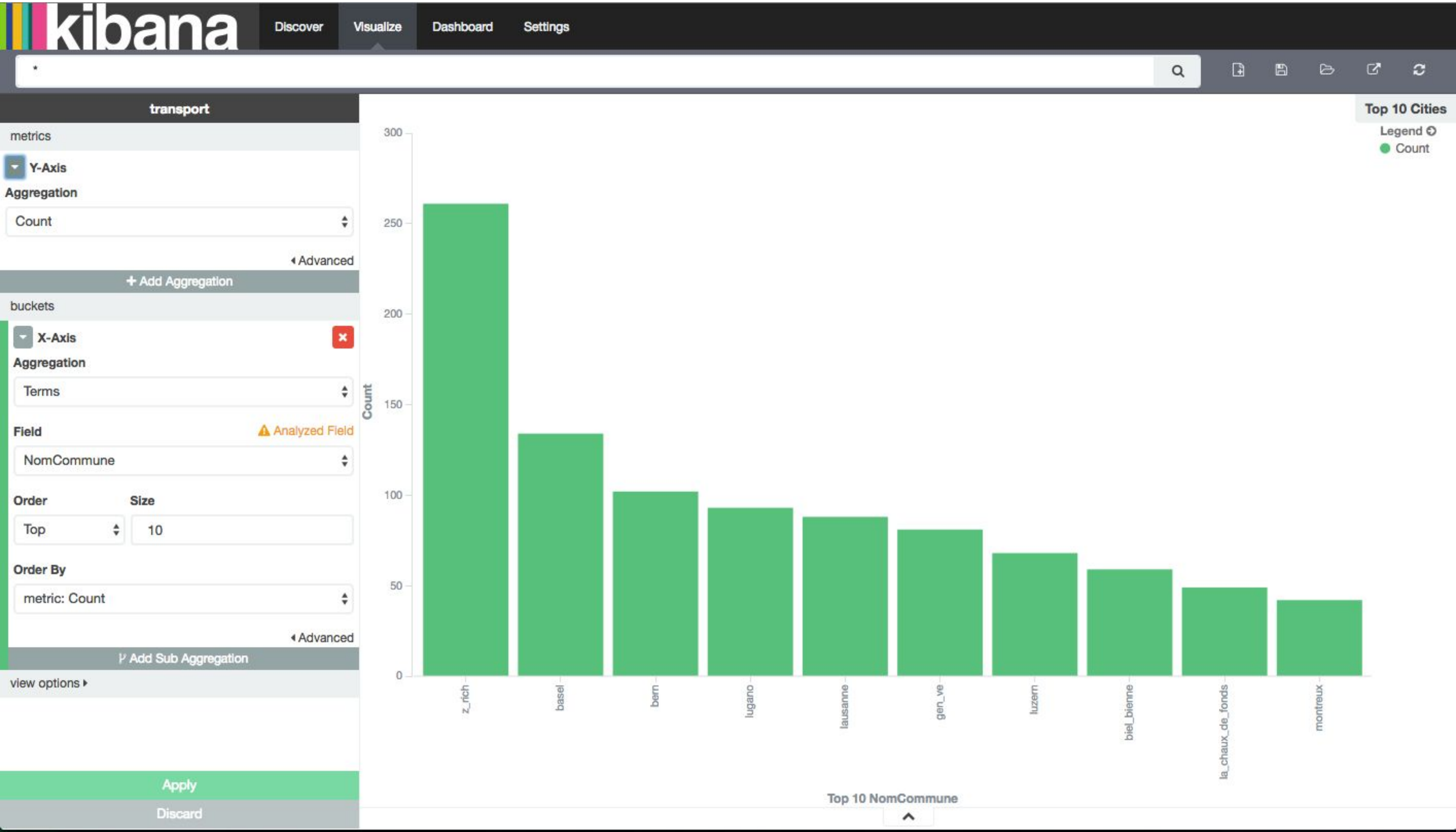
Aggregation: Geohash

Field: location

Precision: 5

kibana
Advanced
Add Sub Aggregation
view options
Apply
Discard
Leaflet | Tiles by MapQuest — Map data © OpenStreetMap contributors, CC-BY-SA

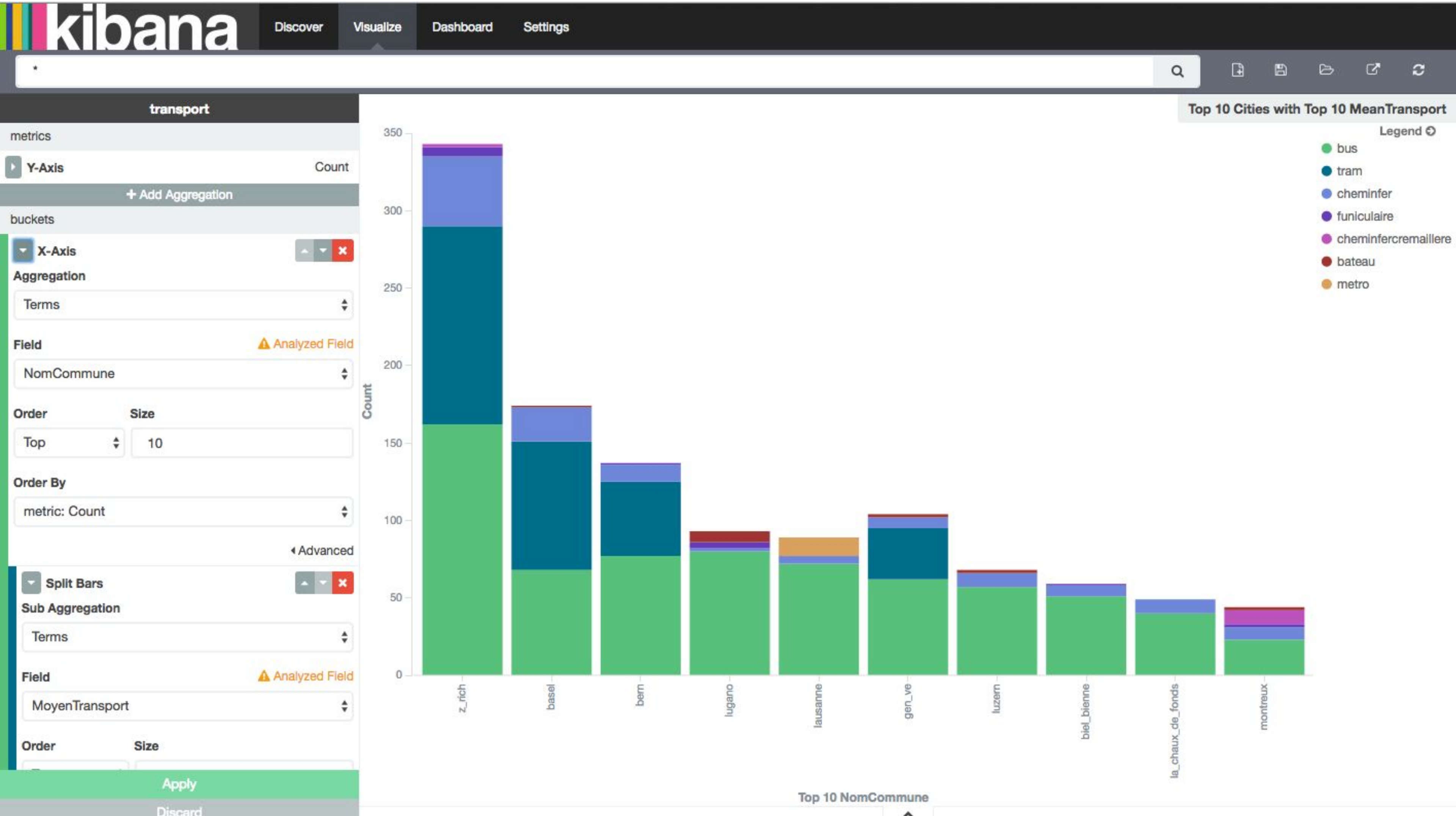
Kibana - Vertical Bar Chart - Top 10 Cities



Y-Axis
Aggregation: Count

X-Axis
Aggregation: Terms
Field: NomCommune
Order: Top
Size: 10
Order By: metric: Count

Kibana - Vertical Bar Chart with Sub-Aggregation - Top 10 Means by Top 10 Cities



transport

metrics

Y-Axis Count

buckets

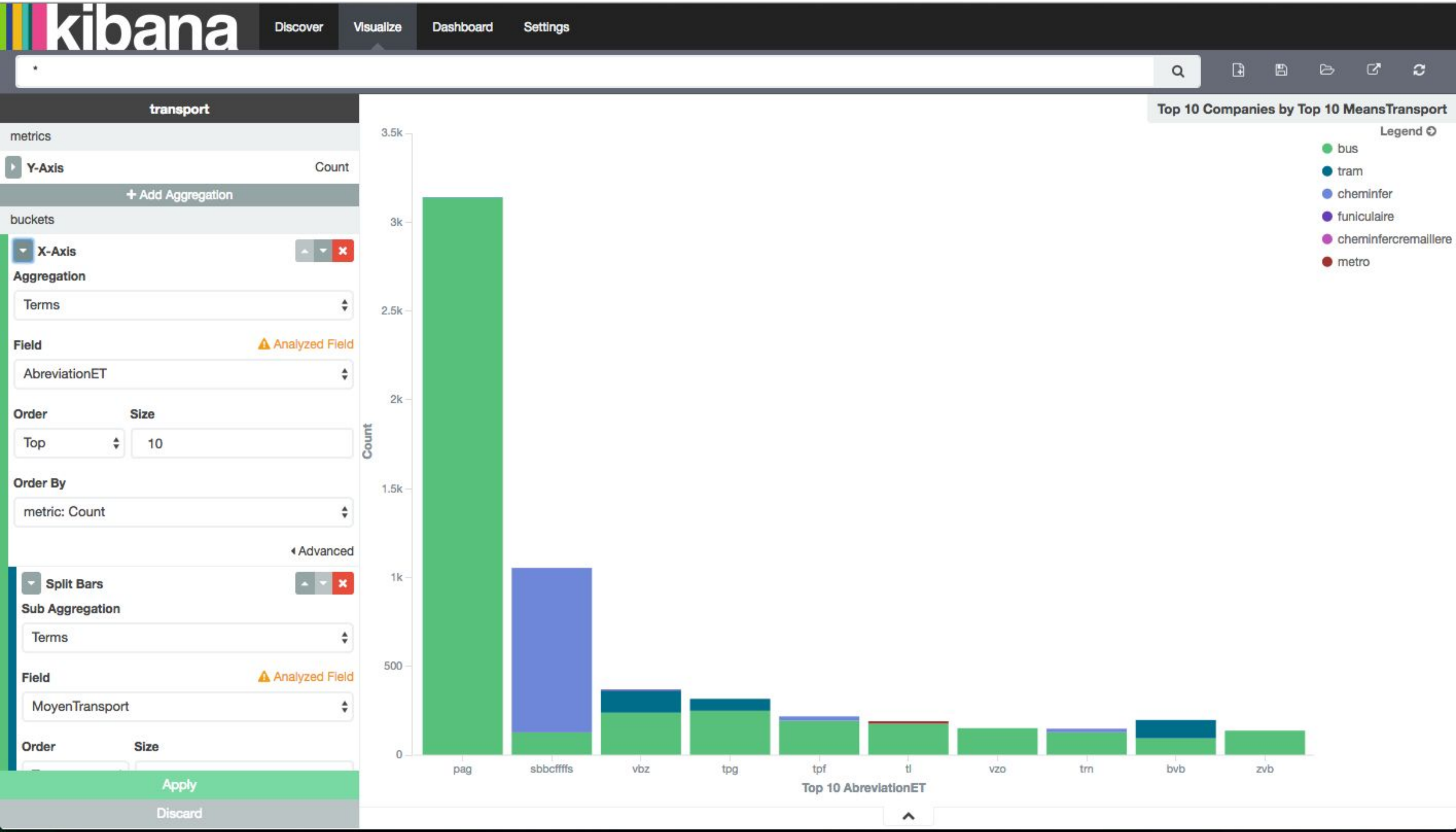
X-Axis
Aggregation: Terms
Field: NomCommune ⚠ Analyzed Field
Order: Top Size: 10
Order By: metric: Count

Split Bars
Sub Aggregation: Terms
Field: MoyenTransport ⚠ Analyzed Field
Order Size

Top 10 Cities with Top 10 MeanTransport

Legend
- ● bus
- ● tram
- ● cheminfer
- ● funiculaire
- ● cheminfercremaillere
- ● bateau
- ● metro

Kibana - Top 10 Means by Top 10 Companies VERSUS Top 10 Companies by Top 10 Means



transport

metrics

Y-Axis: Count

buckets

X-Axis

Aggregation: Terms

Field: AbreviationET (Analyzed Field)

Order: Top

Size: 10

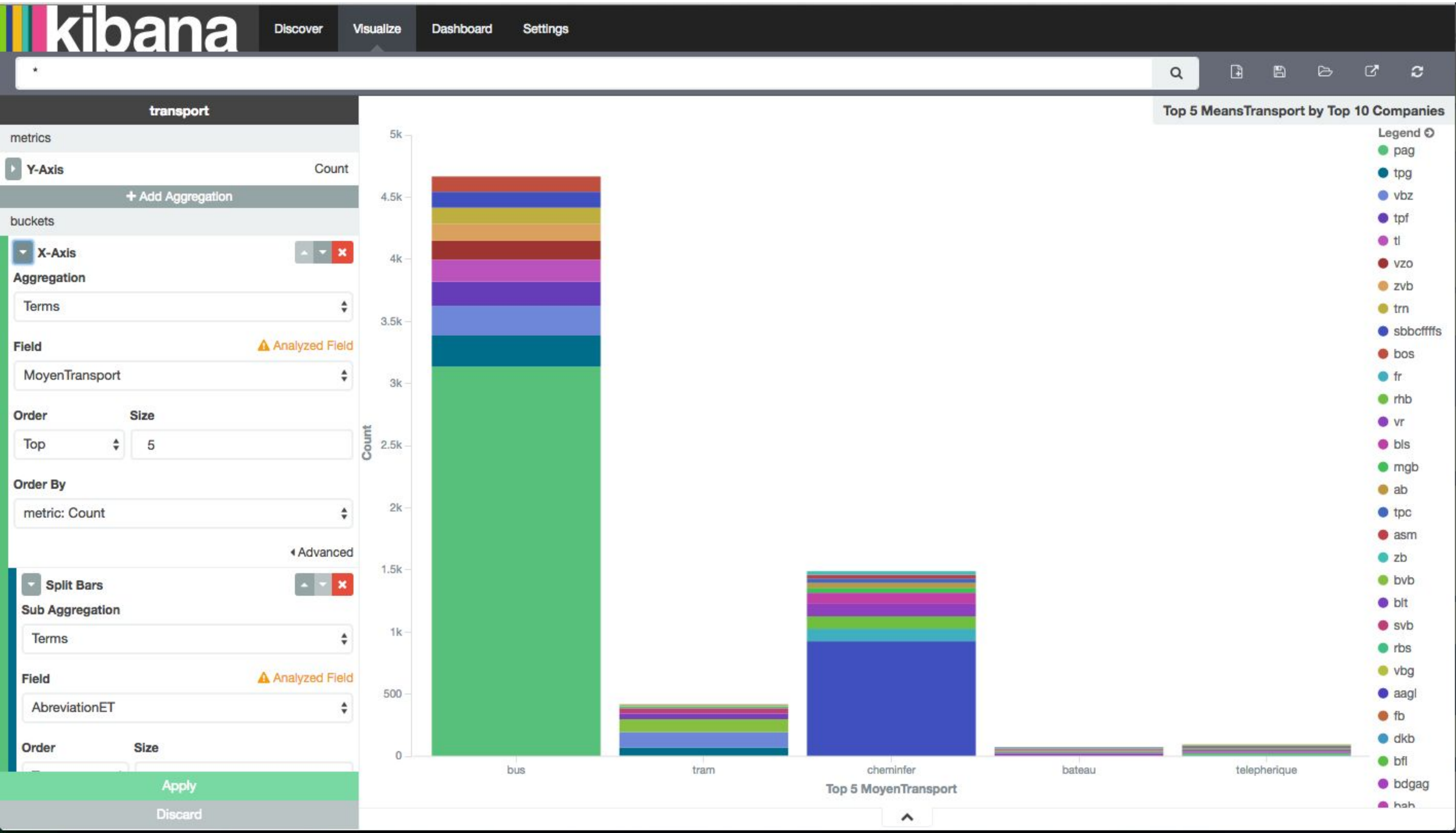
Order By: metric: Count

Split Bars

Sub Aggregation: Terms

Field: MoyenTransport (Analyzed Field)

Kibana - Top 10 Means by Top 10 Companies VERSUS Top 10 Companies by Top 10 Means

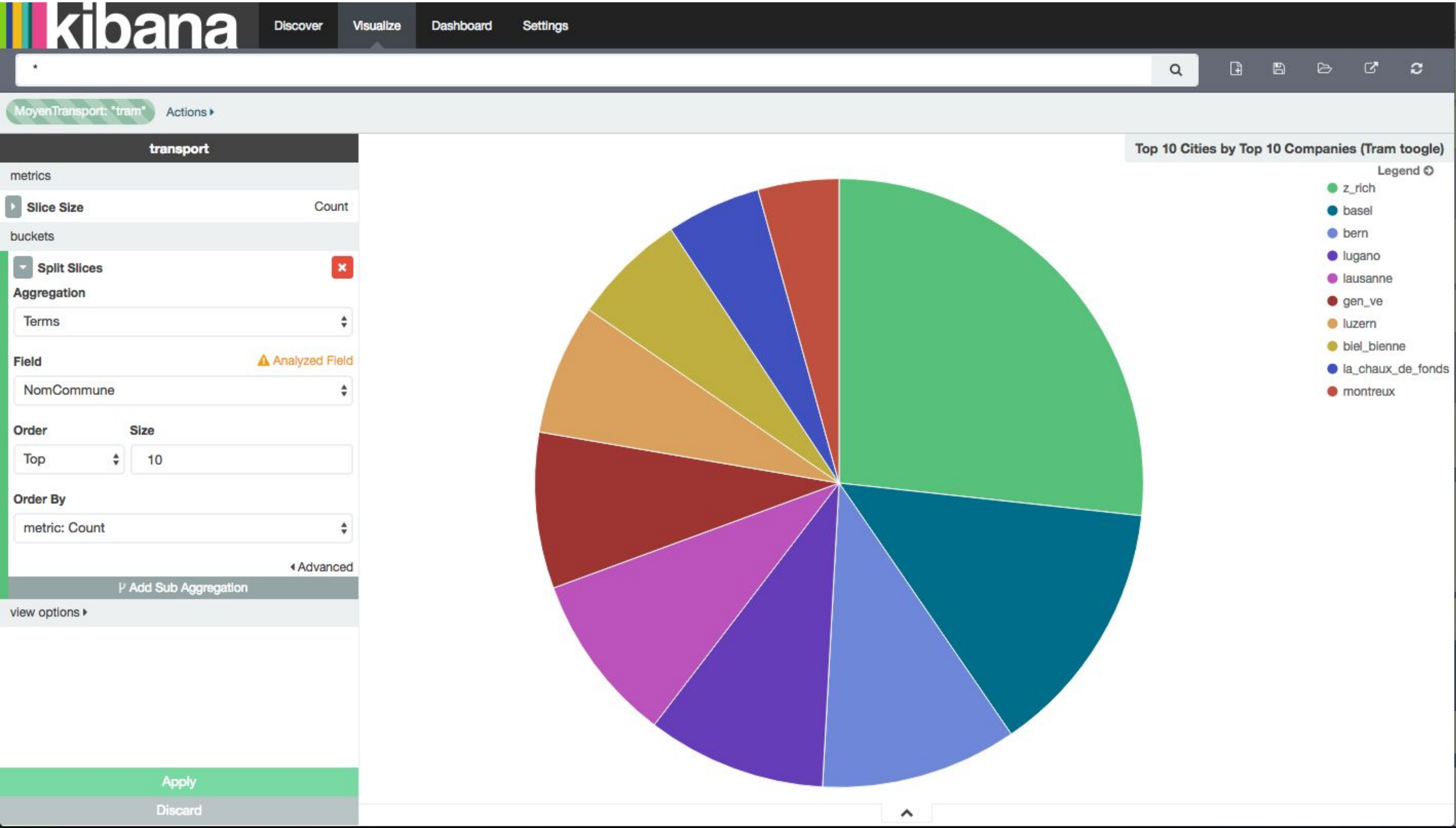


Y-Axis Count

X-Axis
Aggregation: Terms
Field: MoyenTransport (Analyzed Field)
Order: Top
Size: 5
Order By: metric: Count

Split Bars
Sub Aggregation: Terms
Field: AbreviationET (Analyzed Field)
Order
Size

Kibana - Togglable Fields (OFF)



transport

metrics

Slice Size — Count

buckets

Split Slices

Aggregation: Terms

Field: NomCommune — ⚠ Analyzed Field

Order: Top — Size: 10

Order By: metric: Count

Advanced

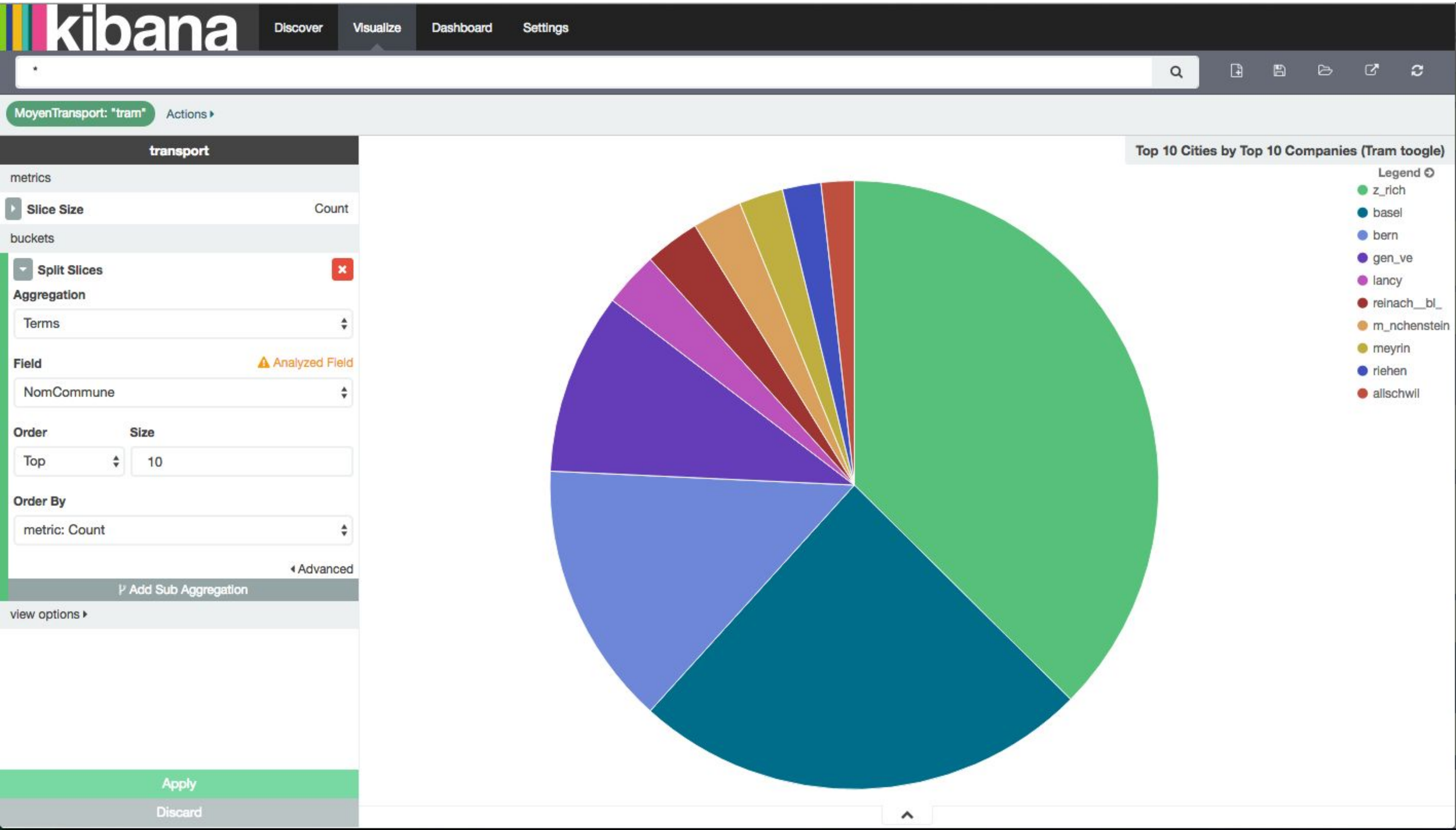
Add Sub Aggregation

view options

Top 10 Cities by Top 10 Companies (Tram toogle)

● z_rich
● basel
● bern
● lugano
● lausanne
● gen_ve
● luzern
● biel_bienne
● la_chaux_de_fonds
● montreux

Kibana - Togglable Fields (ON)



MoyenTransport: "tram"

transport

metrics

Slice Size Count

buckets

Split Slices

Aggregation: Terms

Field: NomCommune ⚠ Analyzed Field

Order: Top Size: 10

Order By: metric: Count

Advanced

Add Sub Aggregation

view options

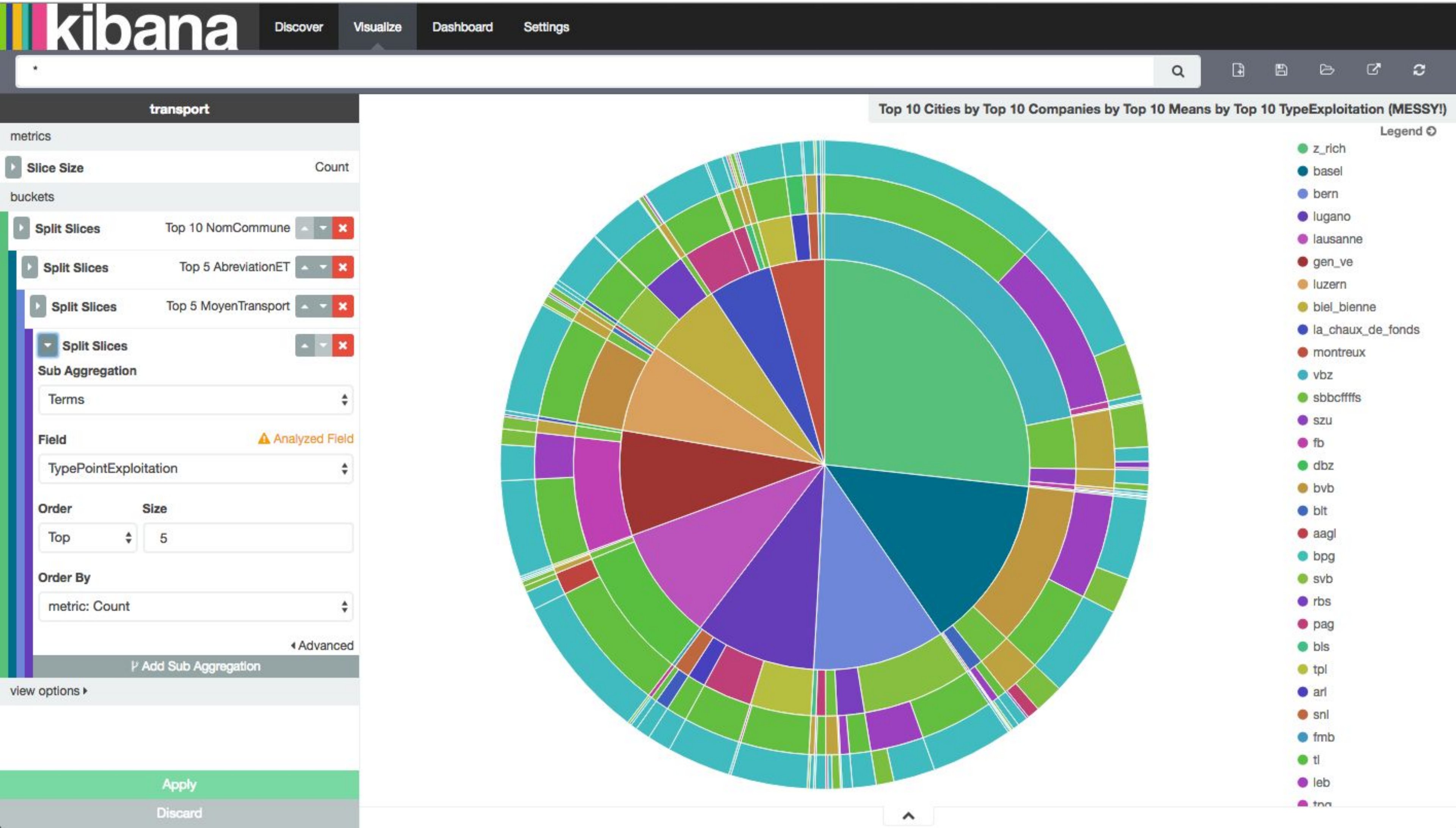
Apply

Discard

Top 10 Cities by Top 10 Companies (Tram toogle)

- ● z_rich
- ● basel
- ● bern
- ● gen_ve
- ● lancy
- ● reinach__bl_
- ● m_nchenstein
- ● meyrin
- ● riehen
- ● allschwil

Kibana - Keep it simple - Don't be messy !



transport

metrics

Slice Size: Count

buckets

Split Slices: Top 10 NomCommune

Split Slices: Top 5 AbreviationET

Split Slices: Top 5 MoyenTransport

Split Slices

Sub Aggregation: Terms

Field: TypePointExploitation (⚠ Analyzed Field)

Order: Top

Size: 5

Order By: metric: Count

Top 10 Cities by Top 10 Companies by Top 10 Means by Top 10 TypeExploitation (MESSY!)

elastic +

Elasticsearch as an AWS Amazon Service ?

When building ELK for your business...

- option 1: deploy ELK in IT: Elastic, Logstash, Kibana = full stack solution
- option 2: use an ELK-as-a-service like Amazon or competitors

http://cloudacademy.com/blog/amazon-elasticsearch-review/

elastic +

Elasticsearch as an AWS Amazon Service ?

Advantages ?

- All logs accessible in one place
- Nothing to install – use any log shipper you want
- Hassle-free managed ELK you don’t need to maintain and scale
- easy to connect my existing services (IT or Amazon) to Amazon ELS
- Use built-in Kibana or use your own Kibana or Grafana with Logsene
- price: pay-as-you-go and per-service (no IT & infrastructure, maintenance...)
- price are low if the logs are from local Amazon source, transfer to or from the internet costs: see next slide.

Elasticsearch as an AWS Amazon Service ? Prices ?

- Machine: the cheapest is t2.micro.elasticsearch, 0.021$/h, 15.12$/month.
- storage: from $0.079/GB/month (79 $/TB/month) on magnetic disk (HDD)
- transfer of large amounts to or from the internet costs. All the log data have to go IN first: it's free from the internet, but might cost from certain services (up to $ 0.01/GB, 10 $/TB). Transferring data OUT to the internet is expensive (up to $0.090 per GB, that's 90$/TB).
- do all other computation also on an Amazon service (with free or cheap transfer rate). Usually Kibana will only transfer small amounts of data for what it needs to render the graphics or the search query.

https://aws.amazon.com/elasticsearch-service/pricing/

elastic +

Elasticsearch as an AWS Amazon Service ?

Drawbacks (in favor of ELK Full-stack in IT dept.)

- version of Elasticsearch: not choosable, and Elasticsearch 1.5.x and other versions have critical bugs
- less flexible with library support, the technology moves fast!
- limited choices in terms of VM characteristics: (type, disk size, RAM)
- requires cloud usage (country and legal issues, technical issues with traffic/bandwidth)

Competitors of ELS on AWS ?

- **Logsene** by SemaText is an alternative service that integrates with SPM Performance Monitoring. Available in the Cloud and On Premise. Correlate logs with performance metrics via SPM. UI view of logs (kibana-like).
- **Whoosh** is developed in Python (ELS in Java). Not as powerful but useful if you code in python.
- **Apache Solr** needs a schema (not ELS), has bigger community. Better performances.

https://sematext.com/logsene/ https://pypi.python.org/pypi/Whoosh/ https://lucene.apache.org/solr/

To go further

Amazon official documentation

https://aws.amazon.com/elasticsearch-service
http://aws.amazon.com/en/documentation/elasticsearch-service/

Elastic.co official documentation

https://www.elastic.co/products/elasticsearch
https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html
http://aws.amazon.com/en/documentation/elasticsearch-service/

Other sources of information

https://wikitech.wikimedia.org/wiki/Logstash
https://cloudacademy.com/amazon-web-services/advanced-techniques-for-aws-monitoring-metrics-and-logging-course/
https://github.com/elastic/elasticsearch
http://www.elasticsearchtutorial.com/elasticsearch-in-5-minutes.html

Thanks! Any questions ?

Elasticsearch, Logstash & Kibana

Search & Analyze log files in clusters & Real Time

Berney L�onard, Bron Sacha, Minder Valentin, Salathe Fabien
HEIG-VD - CLD - 2016

elastic logstash kibana