elastic

Kibana Data Analyst

An Elastic Training Course

7.1.1

elastic.co/training

Kibana Data Analyst

Course: Kibana Data Analyst

Welcome to This Virtual Training

- We will start momentarily
- The training will start with an audio/video test, to make sure that everyone can hear and see the instructors
- To prevent any audio/video issues, please:
  - disable any ad blockers or script blockers
  - use a supported web browser: Chrome or Firefox
- In case of problems, try the following steps in order:
  - refresh this web page
  - open this page in an "incognito" or "private" window
  - try another web browser
  - as a last resort, restarting your computer sometimes helps too

Welcome to This Training

- Visit training.elastic.co and log in
  - **follow instructions from registration email to get access**
- Go to "**My Account**" and click on today's training
- Download the PDF file (this contains all the slides)
- Click on "**Virtual Link**" to access the Lab Environment
  - create an account
  - you will need an access token, which the instructor will provide

About This Training

- Environment
- Introductions
- Code of Conduct (https://www.elastic.co/community/codeofconduct)
- Agenda...

Course Agenda

1 Kibana Fundamentals

2 Kibana Search

3 Kibana Visualizations

4 Kibana Dashboards

5 Kibana Visual Builder

6 Kibana Management

- ● Kibana Fundamentals
- ○ Kibana Search
- ○ Kibana Visualizations
- ○ Kibana Dashboards
- ○ Kibana Visual Builder
- ○ Kibana Management

Module 1

Kibana Fundamentals

Topics

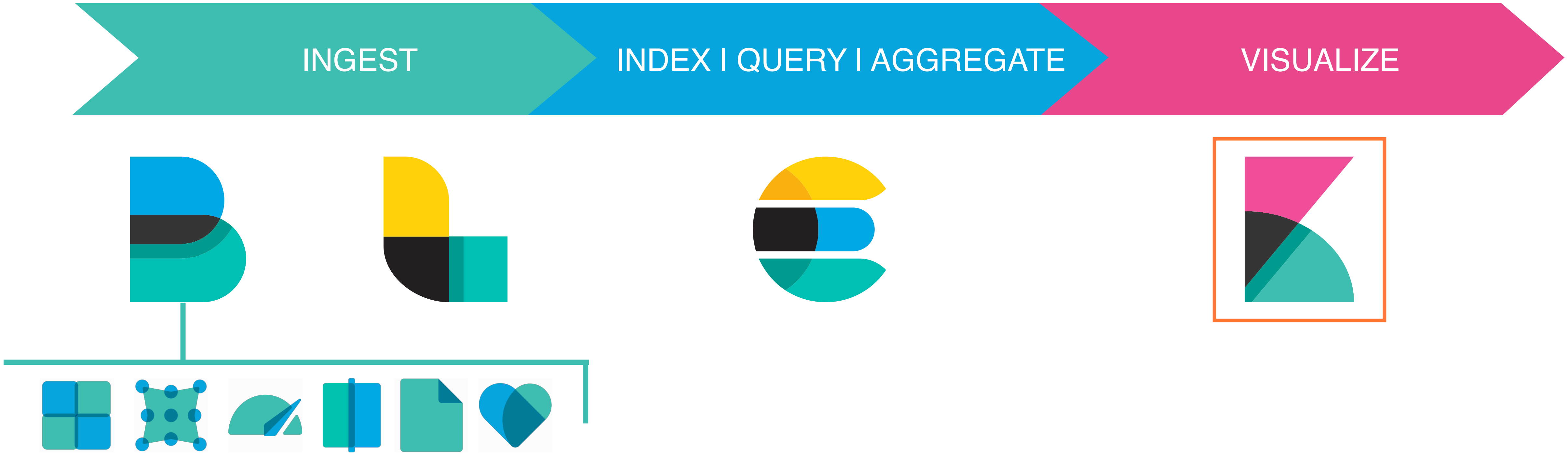
- Introduction to Kibana
- Discover Interface
- Aggregations

Kibana Fundamentals

Lesson 1

Introduction to Kibana

The Elastic Stack



Ingest: Logstash and Beats

- **Logstash**
  - Server-side data processing
  - Ingests data from multiple sources simultaneously (MongoDB, PostgreSQL, Elasticsearch, ...)
  - Parse, transform and prepare your data for ingestion
- **Beats**
  - Single purpose data shippers
  - Many flavors: Filebeat, Metricbeat, Packetbeat, Winlogbeat, ...
  - Lightweight agents that send data from a machine to Elasticsearch or Logstash

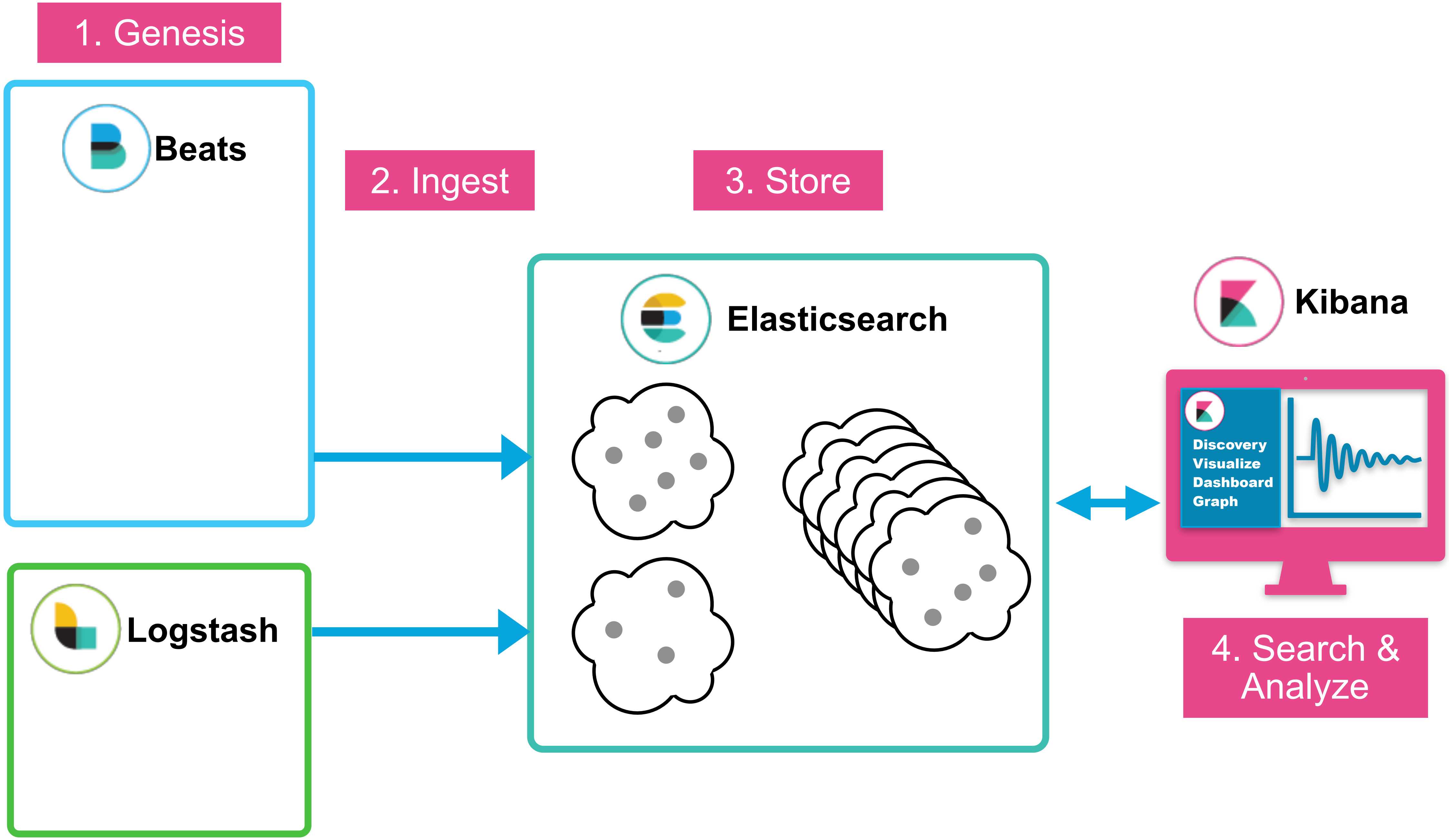
Index: Query and Aggregations

- Elasticsearch
  - Heart of the Elastic Stack
  - **distributed**: easy to scale
  - **RESTful**: easy to communicate with using APIs
  - search, analyze and store data

Visualize

- **Kibana**
  - Window into Elastic Stack
  - Provides Web-based UI to
    - Manage the stack
    - Interact with the data
    - Get data in
    - And more…

Data Journey



Document

- Document
  - Serialized JSON Object
  - Stored in Elasticsearch
  - Has Unique ID

title	category	date	author_first_name	author_last_name	author_company
Fighting Ebola with Elastic	User Stories		Emily	Mosher	

A row in a table

```
{
  "title": "Fighting Ebola with
Elastic",
  "category": "User Stories",
  "author": {
    "first_name": "Emily",
    "last_name": "Mosher"
  }
}
```

JSON

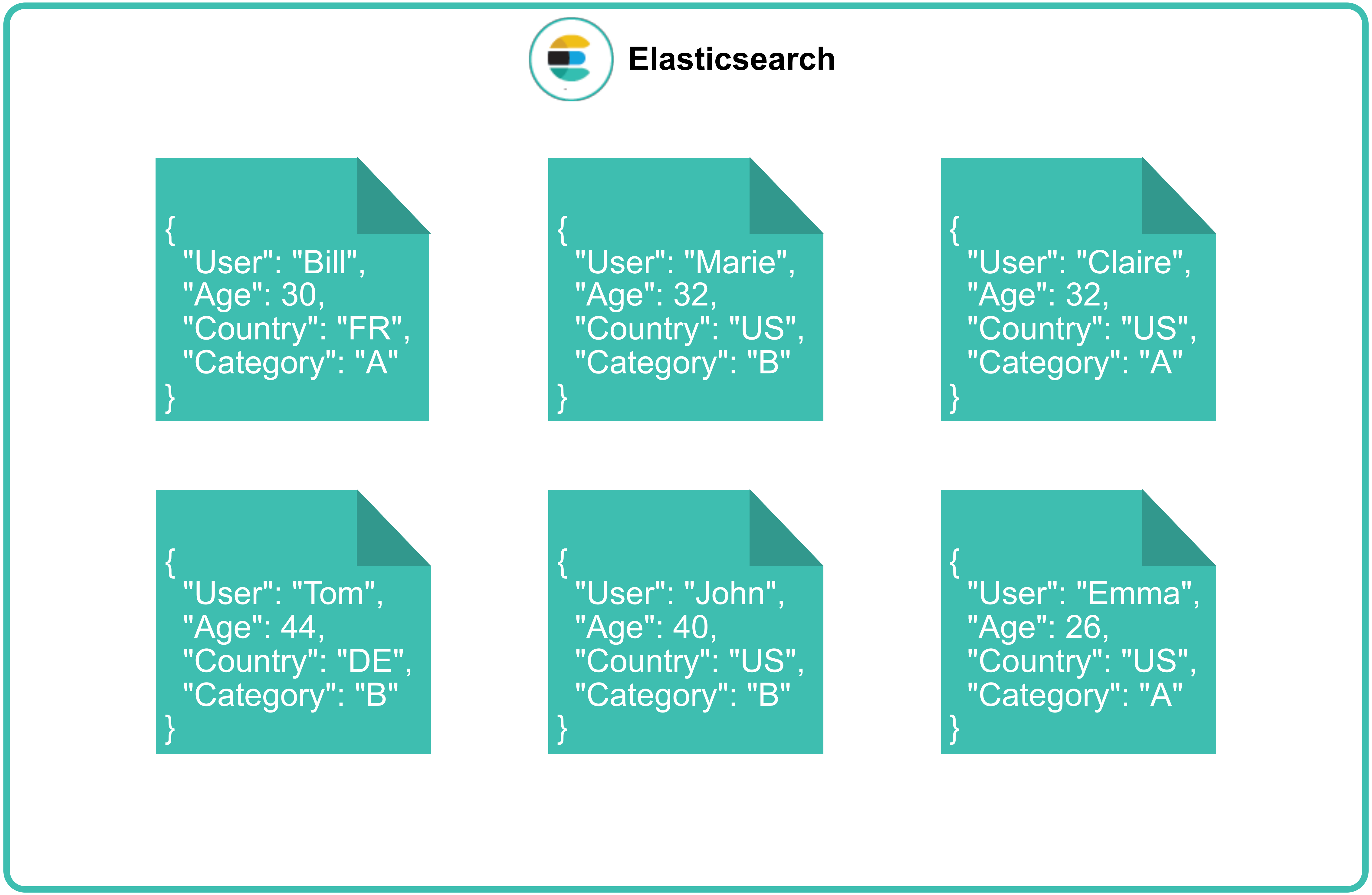
```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <author>
      <first_name>Emily</first_name>
      <last_name>Mosher</last_name>
   </author>
   <category>User Stories</category>
   <title>Fighting Ebola with Elastic</title>
</root>
```

XML

A Simple Example: Spreadsheet

id	user	age	country	category
1	Bill	30	FR	A
2	Marie	32	US	A
3	Claire	32	US	A
4	Tom	44	DE	B
5	John	40	US	B
6	Emma	26	US	B

A Simple Example: Elasticsearch



Elasticsearch

```
{
  "User": "Bill",
  "Age": 30,
  "Country": "FR",
  "Category": "A"
}
```

```
{
  "User": "Marie",
  "Age": 32,
  "Country": "US",
  "Category": "B"
}
```

```
{
  "User": "Claire",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Tom",
  "Age": 44,
  "Country": "DE",
  "Category": "B"
}
```

```
{
  "User": "John",
  "Age": 40,
  "Country": "US",
  "Category": "B"
}
```

```
{
  "User": "Emma",
  "Age": 26,
  "Country": "US",
  "Category": "A"
}
```

Data Categories

- **Time Series Data**
  - Event data associated with a moment in time
  - typically grows rapidly
- **Static Data**:
  - relatively slower growth

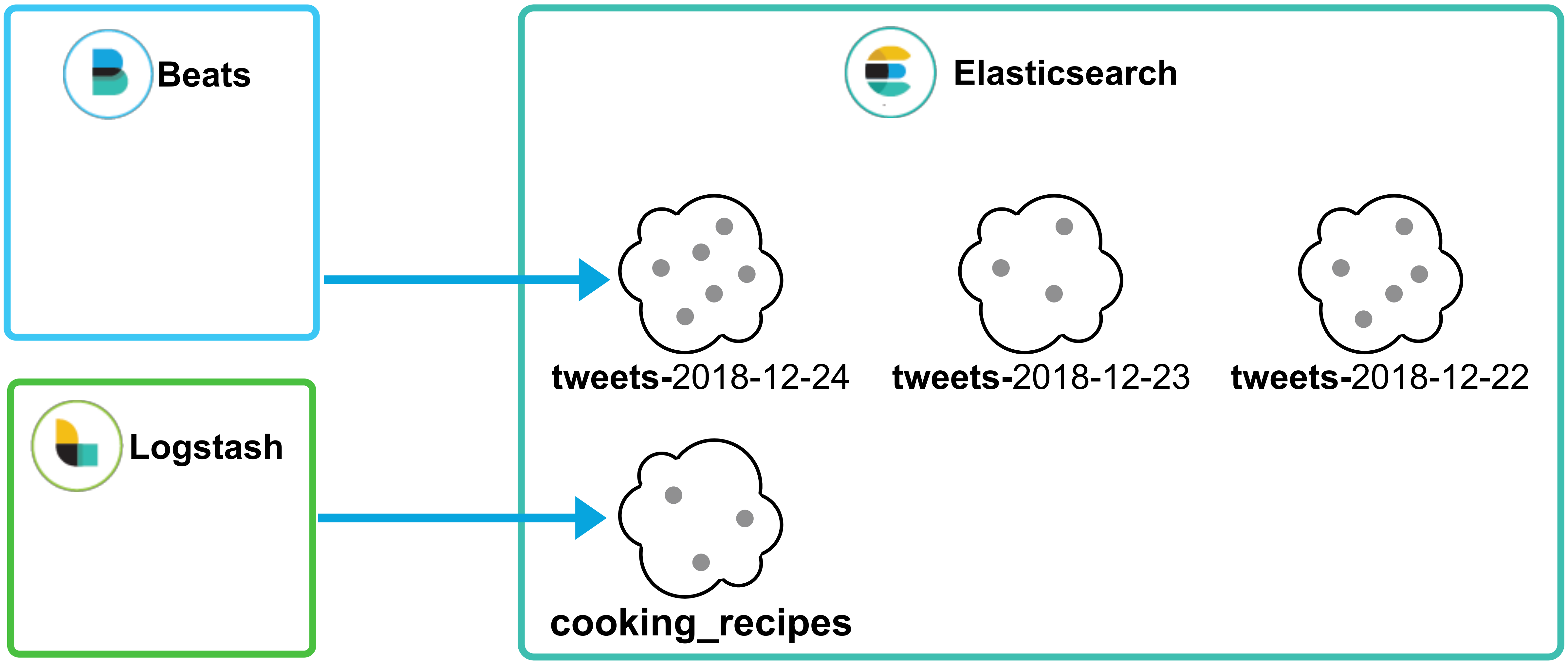
```
{
  "cuisine": "French",
  "ingredients": "Cheese, flour, butter, eggs, milk, nutmeg",
  "time_in_min": 50,
  "level": "easy"
}
```

Which category do these documents belong to?

```
{
  "tweet": "Wow Elasticsearch 7.0 seems awesome!",
  "hashtags": ["elasticsearch", "kibana"]
  "timestamp": September 1st 2017, 07:15:40.035
}
```

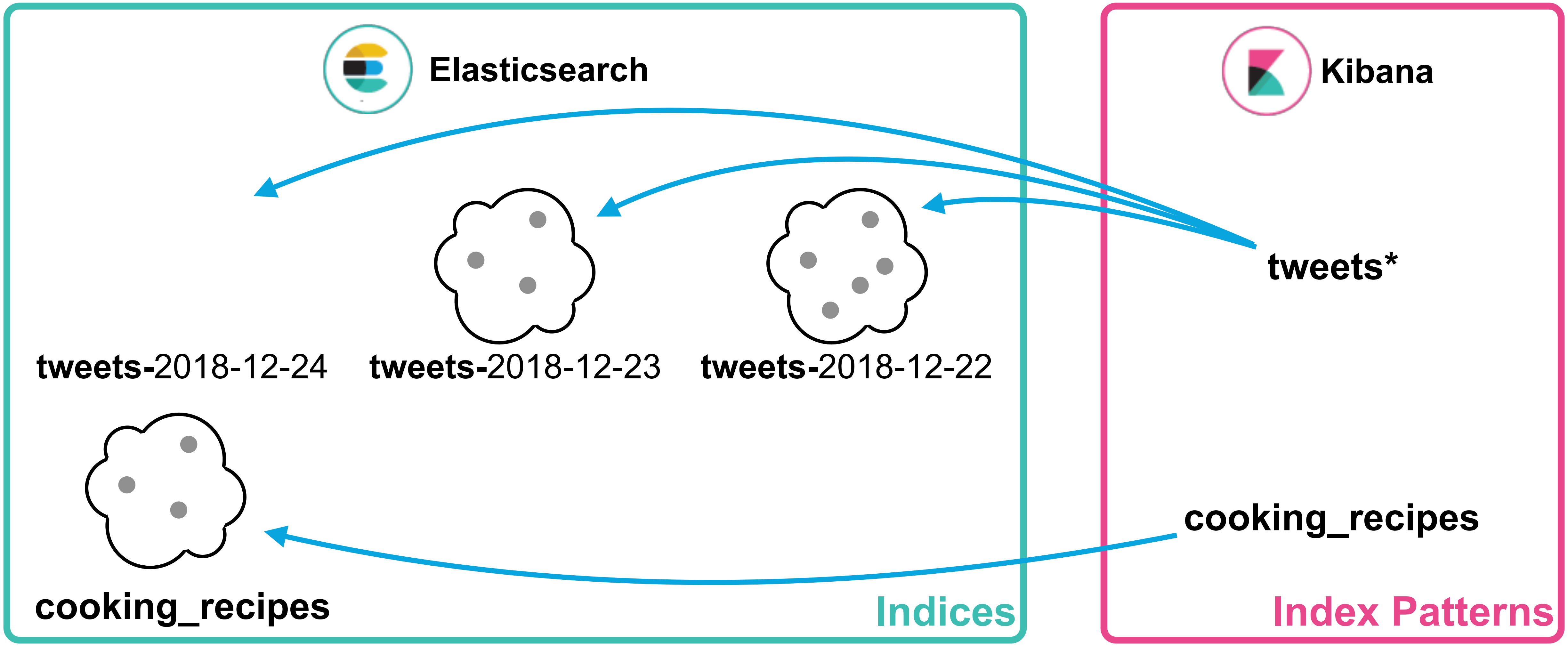
Elasticsearch Index

- Data Container
  - Categorical Index
  - Time Based Index



Kibana Index Pattern

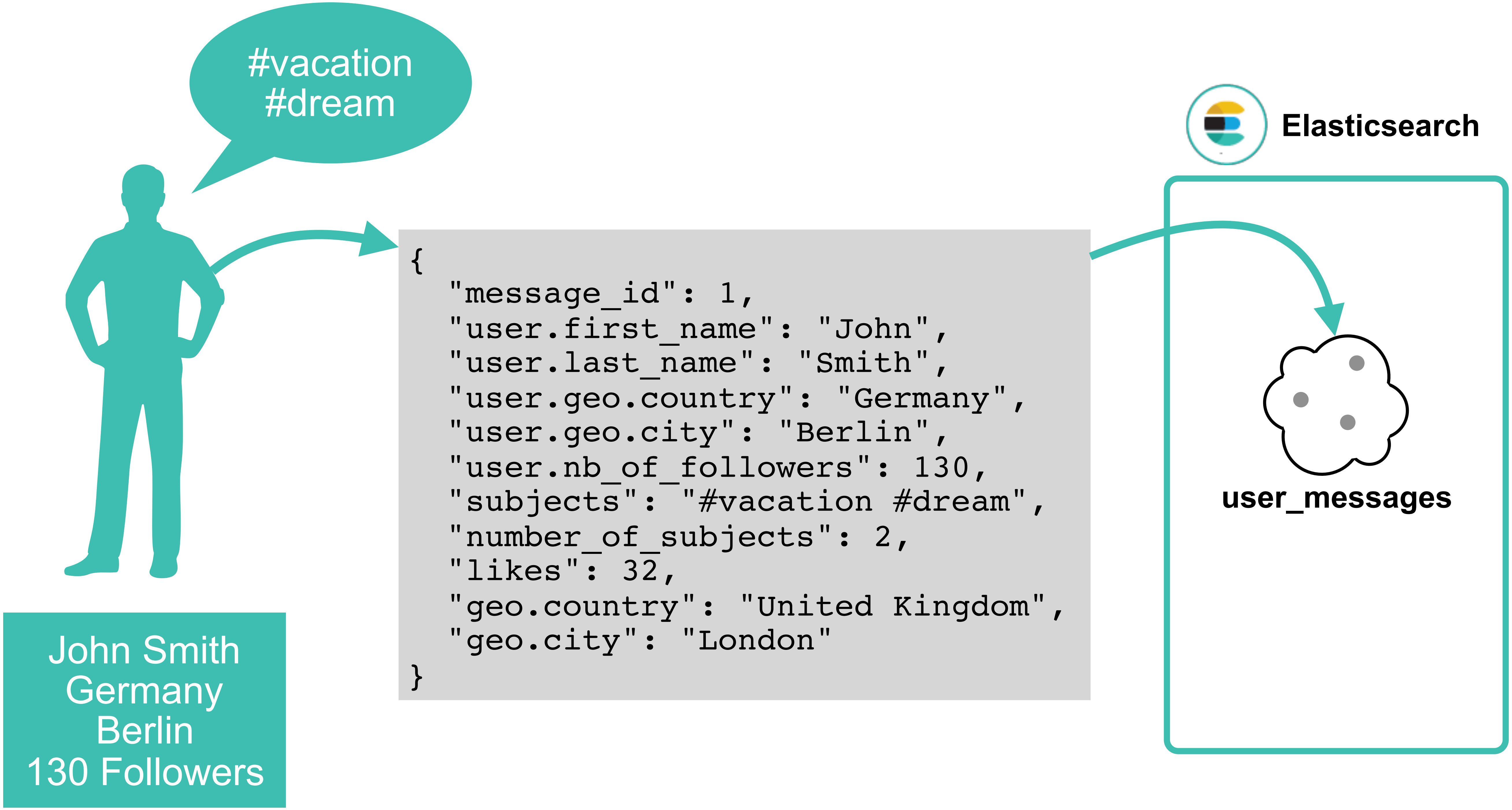
- Points to one or more Elasticsearch **indices**
- Tells Kibana which data you want to work with



Datasets

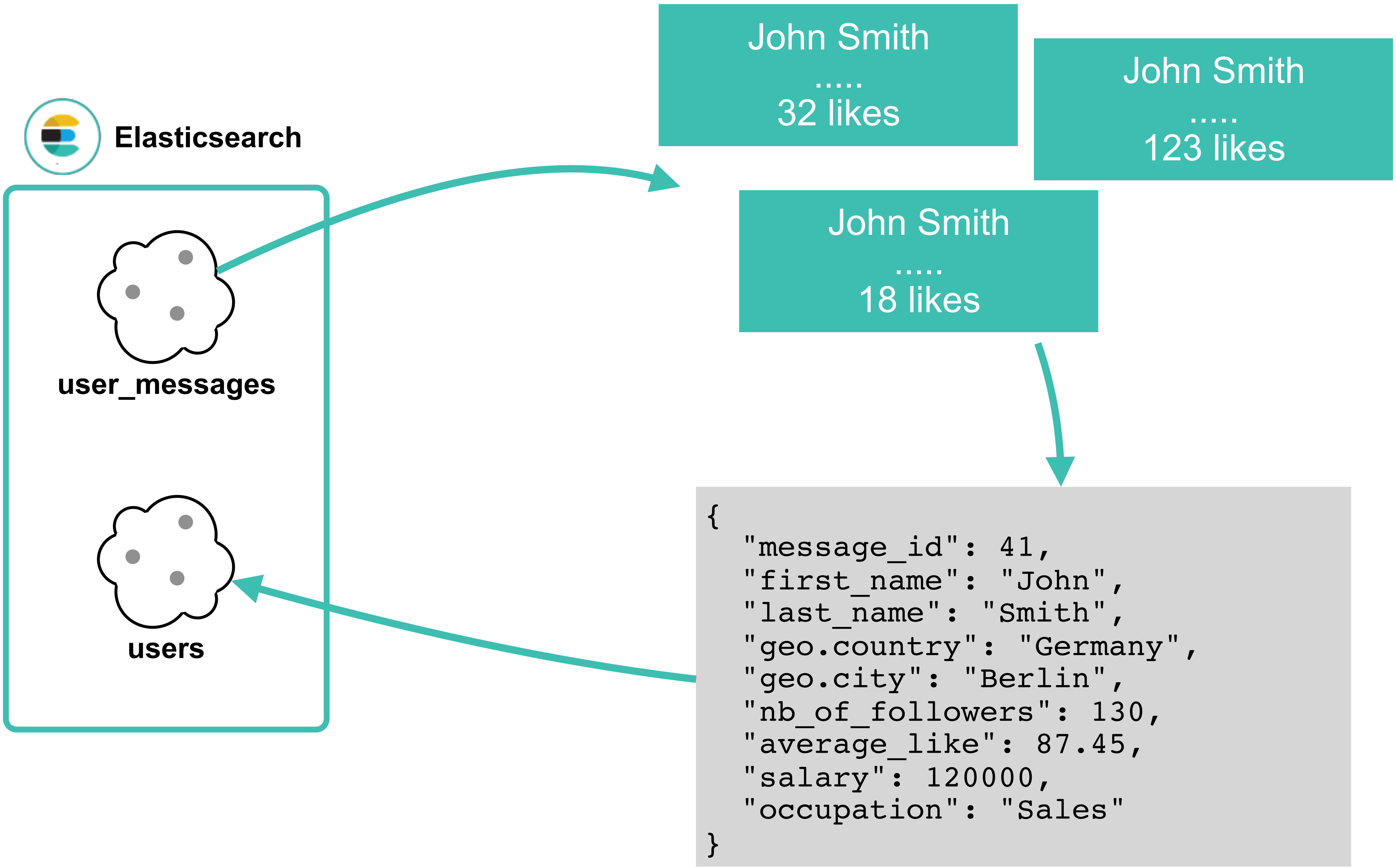
elastic

Messages



```
{
  "message_id": 1,
  "user.first_name": "John",
  "user.last_name": "Smith",
  "user.geo.country": "Germany",
  "user.geo.city": "Berlin",
  "user.nb_of_followers": 130,
  "subjects": "#vacation #dream",
  "number_of_subjects": 2,
  "likes": 32,
  "geo.country": "United Kingdom",
  "geo.city": "London"
}
```

Users



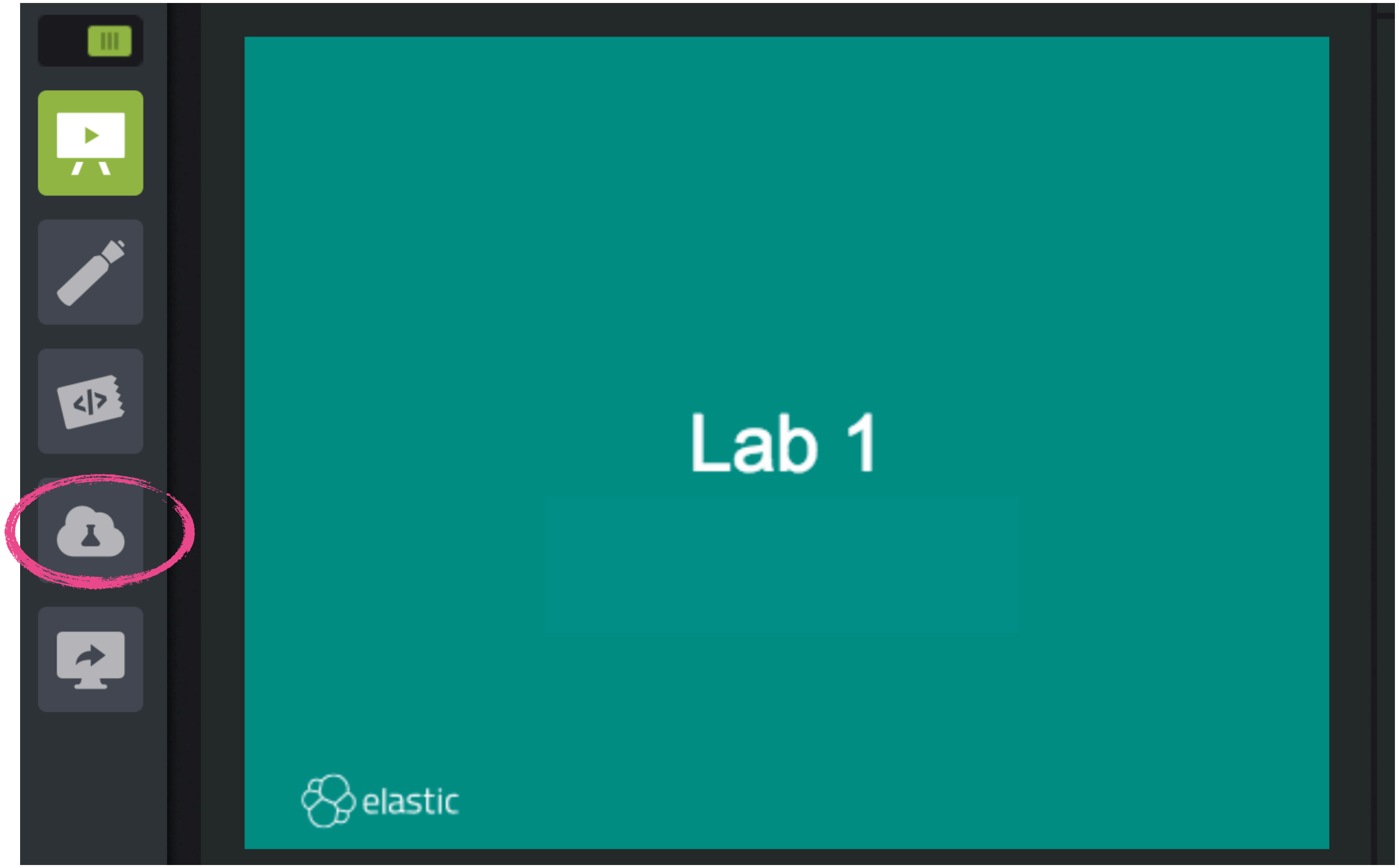
```
{
  "message_id": 41,
  "first_name": "John",
  "last_name": "Smith",
  "geo.country": "Germany",
  "geo.city": "Berlin",
  "nb_of_followers": 130,
  "average_like": 87.45,
  "salary": 120000,
  "occupation": "Sales"
}
```

Lab Environment

elastic

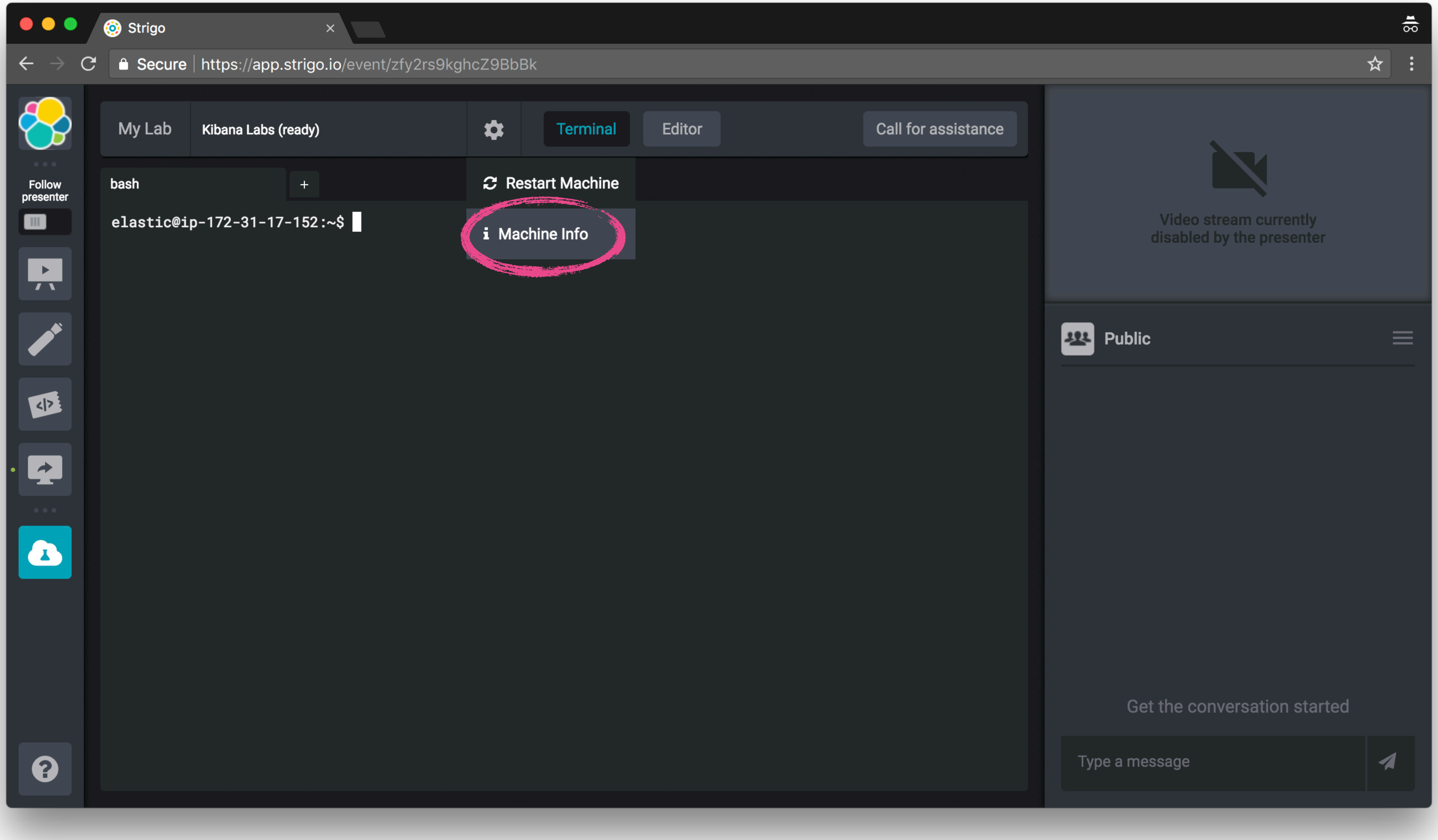
Lab Environment

- Visit Strigo using the link that was shared with you, and log in if you haven't already done so
- Click on "**My Lab**" on the left



Lab Environment

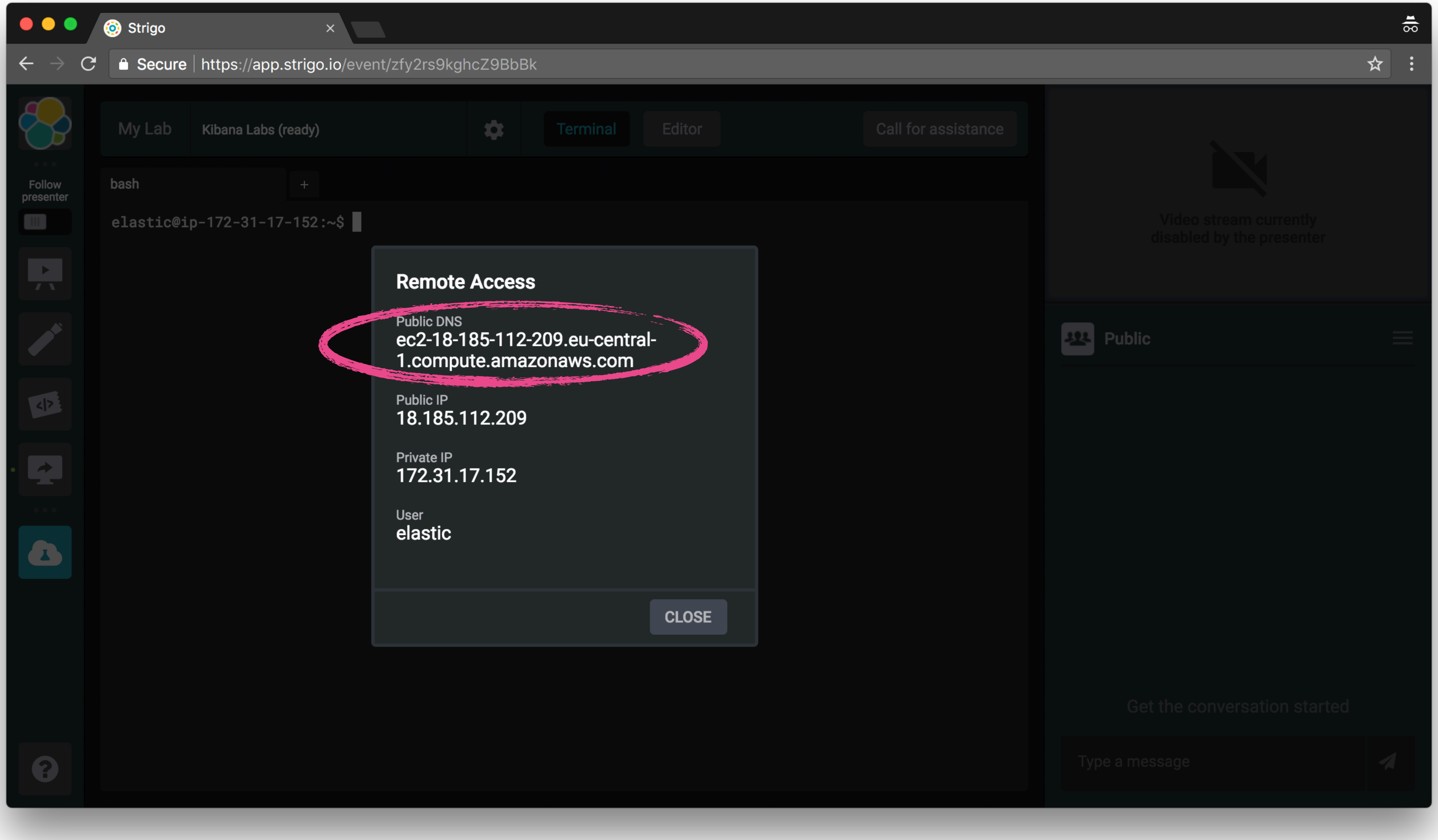
- Click on the gear icon next to "**My Lab**" and select "**Machine Info**"



```
elastic@ip-172-31-17-152:~$
```

Lab Environment

- Copy the hostname that is shown under "**Public DNS**"



Strigo
Secure https://app.strigo.io/event/zfy2rs9kghcZ9BbBk
My Lab
Kibana Labs (ready)
Terminal
Editor
Call for assistance
Follow presenter
bash
Video stream currently disabled by the presenter
Public
Get the conversation started
Type a message
CLOSE

```
elastic@ip-172-31-17-152:~$
```

Remote Access

Public DNS
ec2-18-185-112-209.eu-central-1.compute.amazonaws.com

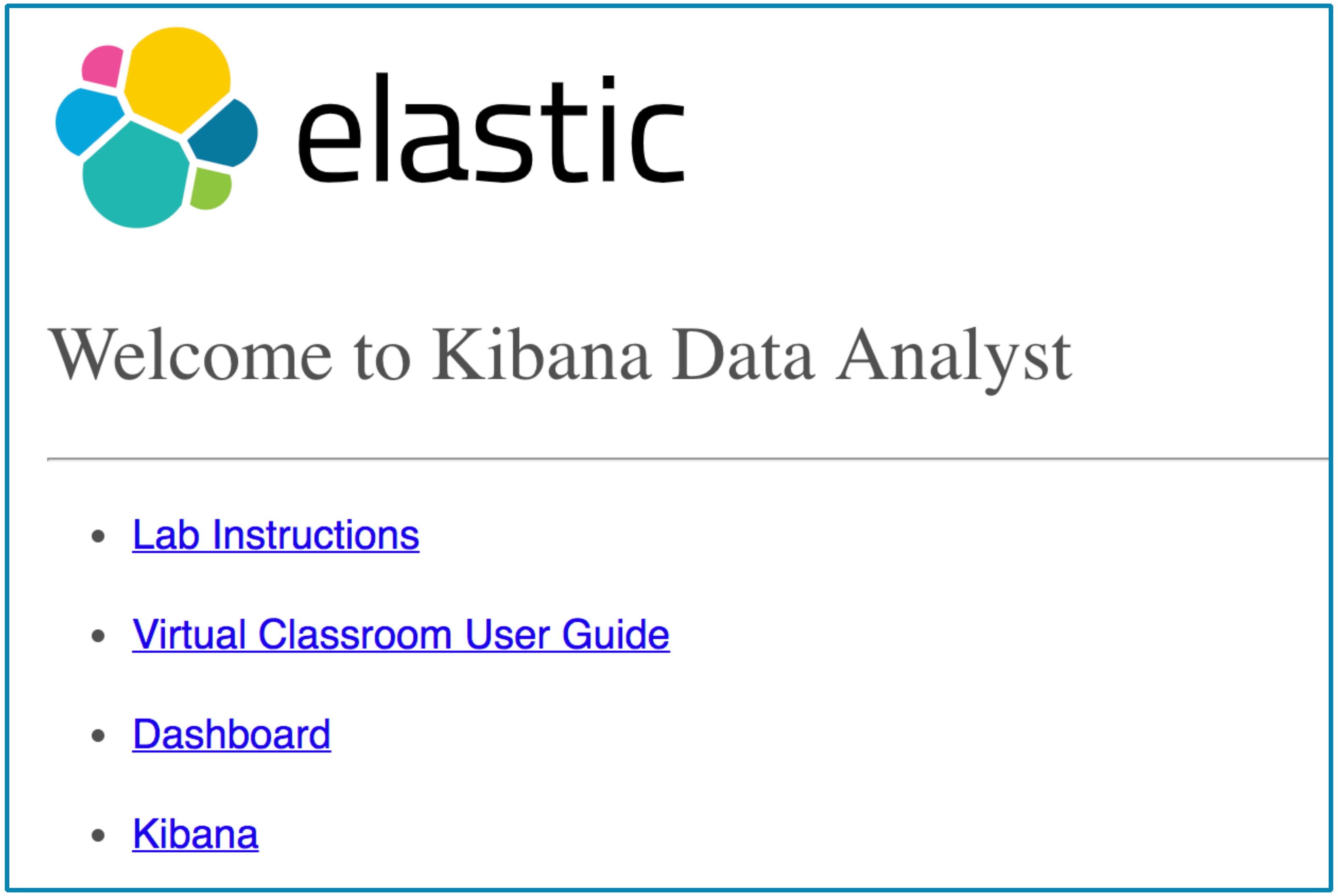
Public IP
18.185.112.209

Private IP
172.31.17.152

User
elastic

Lab Environment

- From here you can access lab instructions and guides
  - you also have them in your .zip file, but it is easier to access and use the lab instructions from here:



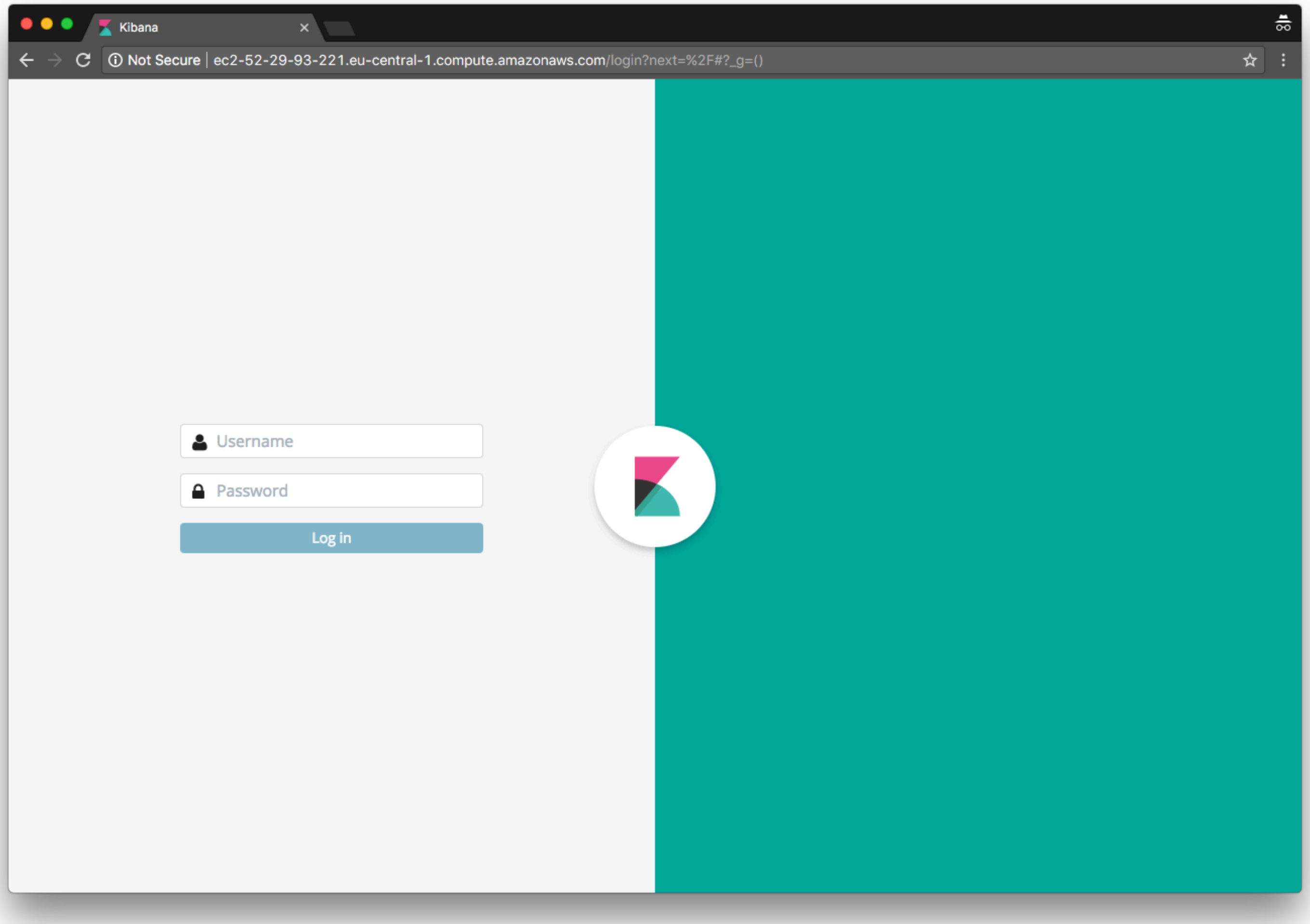
elastic

Welcome to Kibana Data Analyst

- Lab Instructions
- Virtual Classroom User Guide
- Dashboard
- Kibana

Accessing your Cluster

- Click on the Kibana link:
- Log in
  - username: **training**
  - password: **kibana_management**



Kibana
Not Secure | ec2-52-29-93-221.eu-central-1.compute.amazonaws.com/login?next=%2F#?_g=()
Username
Password
Log in

Lesson 1

Review - Introduction to Kibana

elastic

Summary

- Elasticsearch, Kibana, Logstash, and Beats are components of the Elastic Stack
- Kibana can be used to analyze, search, interact with and visualize the data in Elasticsearch
- Kibana can be used to manage the Elastic Stack
- Data is sent as JSON objects into Elasticsearch
- In Kibana, an index pattern can be created to target a specific set of indices

Quiz

1. What are the four main components of the Elastic Stack?
2. **True or False**: Data is stored inside Kibana.
3. What would be a suitable index pattern for accessing both **cooking_recipes** and **cooking_user** indices?
4. What kind of dataset the two following documents belong to?

```
{
  "heartbeat": 123,
  "timestamp": "Mon, 24 Dec 2018 00:23:28 GMT"
}
```

```
{
  "first_name": "Bill",
  "last_name": "Smith",
  "age": 27,
  "country": "Mongolia"
}
```

Lesson 1

Lab - Introduction to Kibana

Kibana Fundamentals

Lesson 2

Discover Interface

elastic

Overview

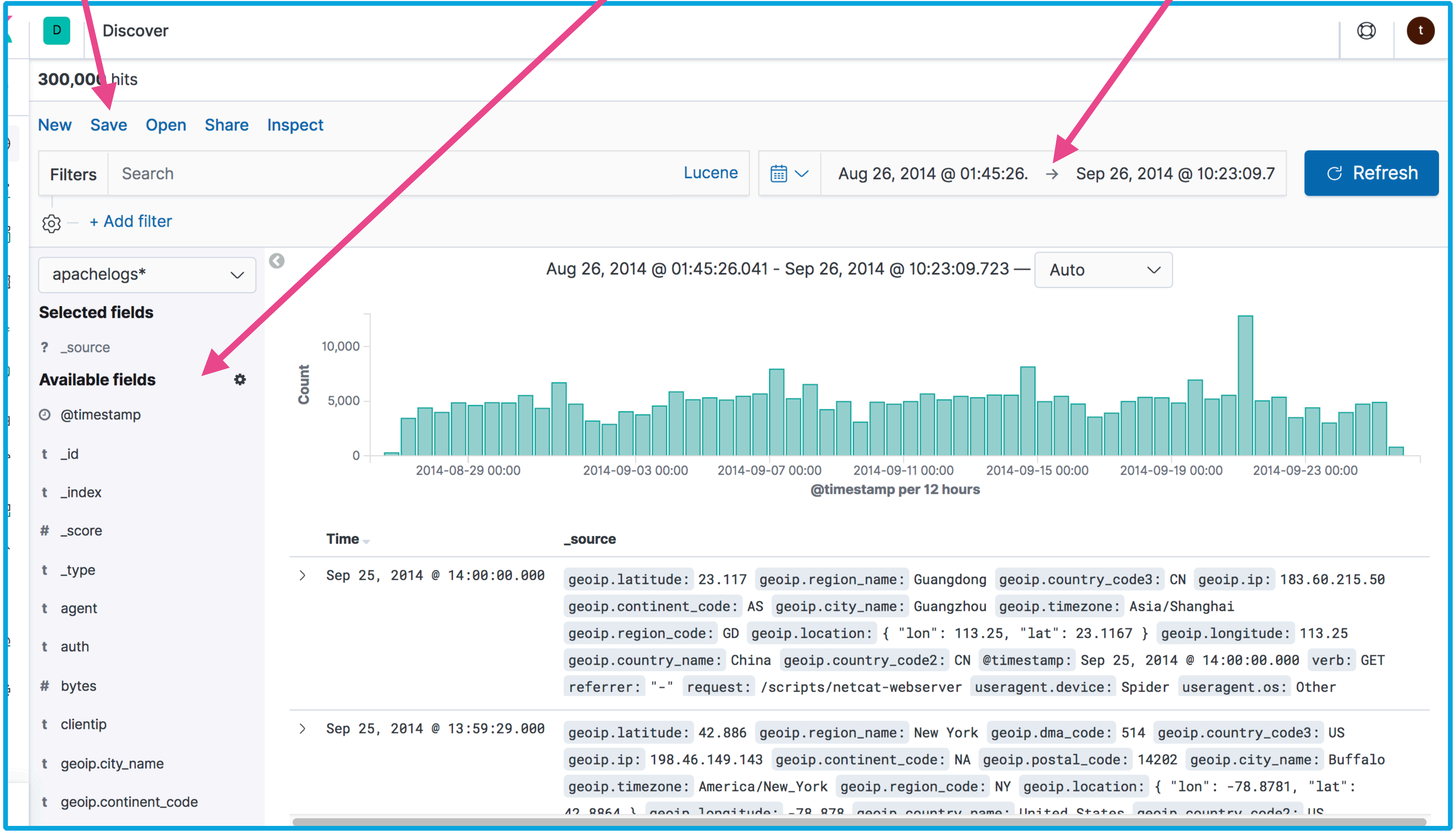
- Elasticsearch data types:
  - numeric
  - text
  - date
  - keywords
  - ...
- Discover interface
  - Explore data in Elasticsearch
  - Slice and Dice (Analyze) Data

Discover Interface

Tool bar

Side navigation

Time picker



300,000 hits

Aug 26, 2014 @ 01:45:26. → Sep 26, 2014 @ 10:23:09.7

Aug 26, 2014 @ 01:45:26.041 - Sep 26, 2014 @ 10:23:09.723 —

Selected fields

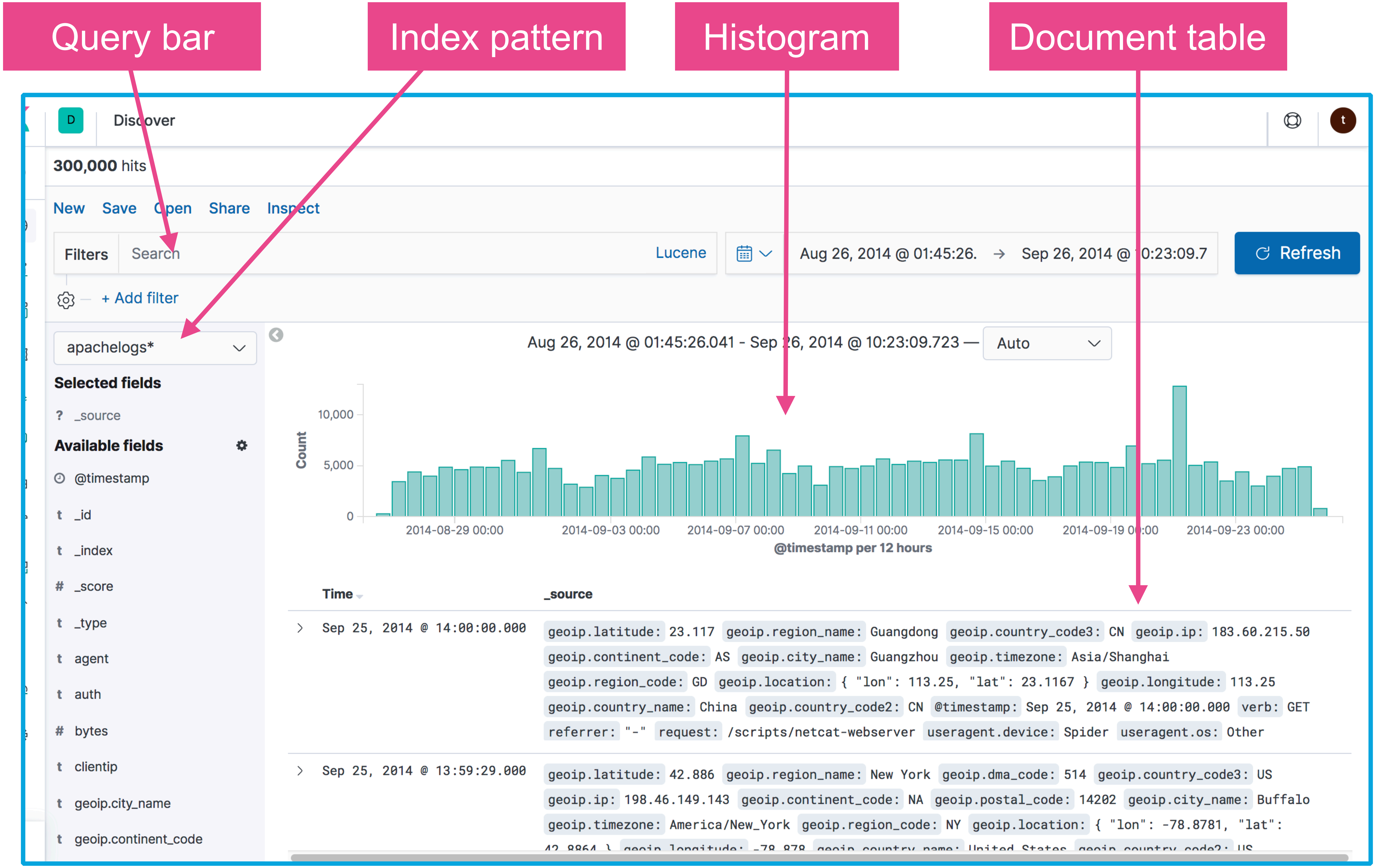
- ? _source

Available fields

- @timestamp
- t _id
- t _index
- # _score
- t _type
- t agent
- t auth
- # bytes
- t clientip
- t geoip.city_name
- t geoip.continent_code

Time	_source
Sep 25, 2014 @ 14:00:00.000	geoip.latitude: 23.117 geoip.region_name: Guangdong geoip.country_code3: CN geoip.ip: 183.60.215.50 geoip.continent_code: AS geoip.city_name: Guangzhou geoip.timezone: Asia/Shanghai geoip.region_code: GD geoip.location: { "lon": 113.25, "lat": 23.1167 } geoip.longitude: 113.25 geoip.country_name: China geoip.country_code2: CN @timestamp: Sep 25, 2014 @ 14:00:00.000 verb: GET referrer: "-" request: /scripts/netcat-webserver useragent.device: Spider useragent.os: Other
Sep 25, 2014 @ 13:59:29.000	geoip.latitude: 42.886 geoip.region_name: New York geoip.dma_code: 514 geoip.country_code3: US geoip.ip: 198.46.149.143 geoip.continent_code: NA geoip.postal_code: 14202 geoip.city_name: Buffalo geoip.timezone: America/New_York geoip.region_code: NY geoip.location: { "lon": -78.8781, "lat": 42.8864 } geoip.longitude: -78.878 geoip.country_name: United States geoip.country_code2: US

Discover Interface



Query bar

Index pattern

Histogram

Document table

Time	_source
Sep 25, 2014 @ 14:00:00.000	geoip.latitude: 23.117 geoip.region_name: Guangdong geoip.country_code3: CN geoip.ip: 183.60.215.50 geoip.continent_code: AS geoip.city_name: Guangzhou geoip.timezone: Asia/Shanghai geoip.region_code: GD geoip.location: { "lon": 113.25, "lat": 23.1167 } geoip.longitude: 113.25 geoip.country_name: China geoip.country_code2: CN @timestamp: Sep 25, 2014 @ 14:00:00.000 verb: GET referrer: "-" request: /scripts/netcat-webserver useragent.device: Spider useragent.os: Other
Sep 25, 2014 @ 13:59:29.000	geoip.latitude: 42.886 geoip.region_name: New York geoip.dma_code: 514 geoip.country_code3: US geoip.ip: 198.46.149.143 geoip.continent_code: NA geoip.postal_code: 14202 geoip.city_name: Buffalo geoip.timezone: America/New_York geoip.region_code: NY geoip.location: { "lon": -78.8781, "lat": 42.8864 } geoip.longitude: -78.878 geoip.country_name: United States geoip.country_code2: US

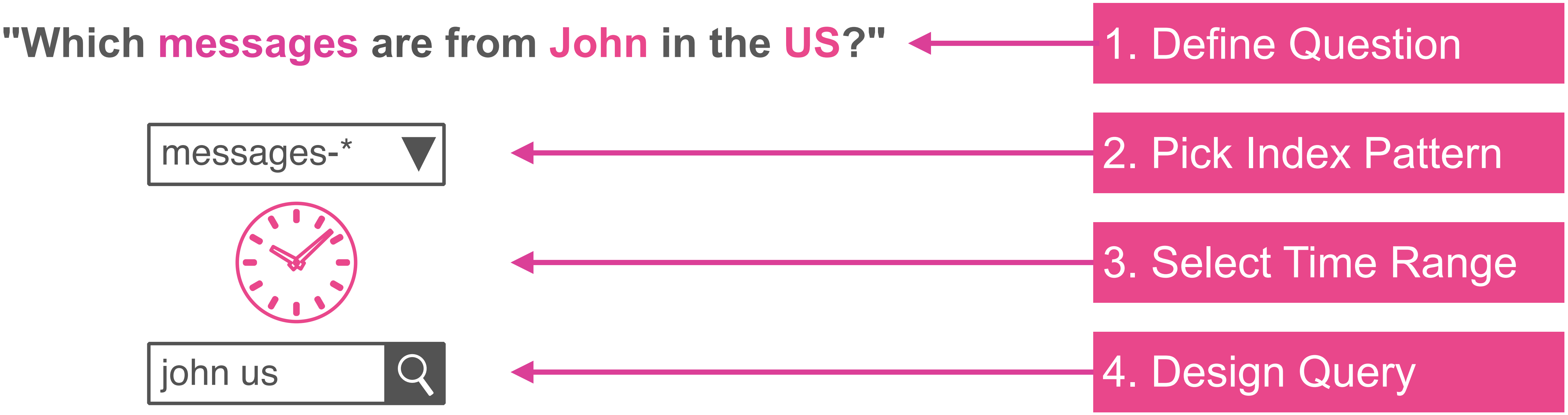
Search is Everywhere

- Elasticsearch is a search engine
  - Kibana can be used to search documents in Elasticsearch
- A search is executed by sending a **query** to Elasticsearch
  - A query can answer many different types of **questions**:
    - who are the users that are called Melissa?
    - what are the names of the people living in France?
    - are there any messages about Netflix?
- In Kibana, a search can be executed from the **query bar**
  - Kibana supports multiple query languages

*

Querying

- Kibana supports multiple query languages



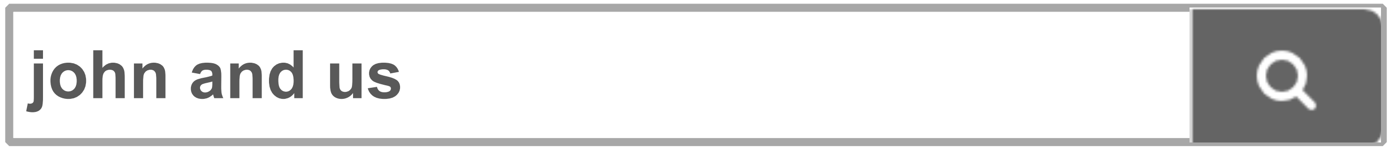
"Which **messages** are from **John** in the **US**?"

1. Define Question
2. Pick Index Pattern
3. Select Time Range
4. Design Query

	id	user	age	country	category
✗	1	Bill	30	FR	A
✓	2	Marie	32	US	A
✓	3	Claire	32	US	A
✓	4	John	40	DE	B
✓	5	John	44	US	B
✓	6	Emma	44	US	B

Search a Specific Field

- By default, the query below will search all **fields** for all values

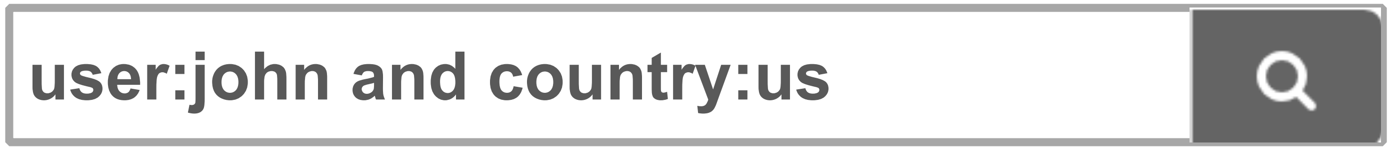


john and us

  - but being more specific will improve search

What are the messages published by user John from country US?

- Query above can be made more specific like this



user:john and country:us

  - Elasticsearch will only need to search limited **fields**

Boolean Operators

- By default, Kibana uses the **or** logic
  - so it matches any documents containing *john* **or** *us*
- Kibana allows you to use the following **boolean operators**:
  - **and**, **or**, and **not**
- Now, you can rewrite the query with the and logic

```
user:john and country:us
```

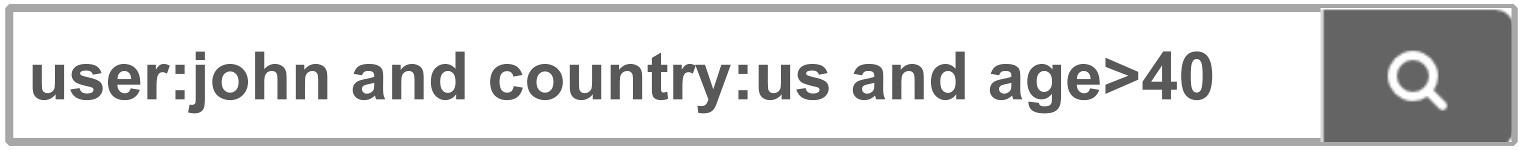
	id	user	age	country	category
✗	1	Bill	30	FR	A
✗	2	Marie	32	US	A
✗	3	Claire	32	US	A
✗	4	John	40	DE	B
✓	5	John	44	US	B
✗	6	Emma	44	US	B

Querying Numeric Fields

- Let's add some complexity to the question:

  What are the messages in which the user is John in the US country whose age is over 40?

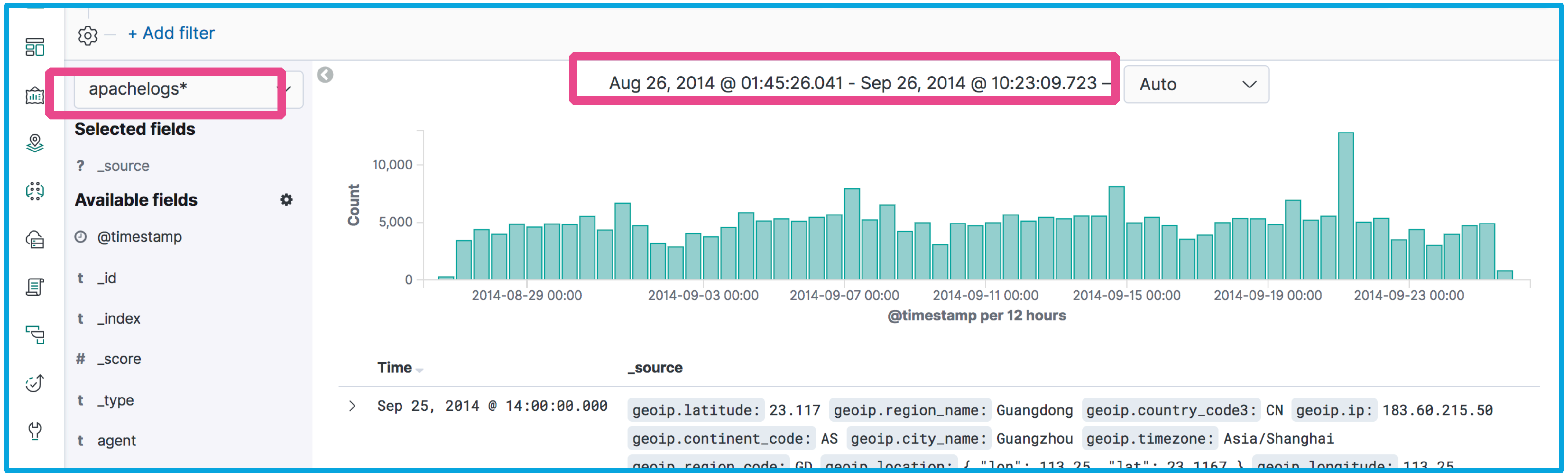
- Numbers are different than text
  - instead of exact matches you often have relations:
    - less than (<)
    - less than or equal (<=)
    - greater than (>)
    - greater than or equal (>=)
- Now, you can rewrite the query as:



```
user:john and country:us and age>40
```

Query "Context"

- Query includes criteria about where to search based on
  - Distribution in Elasticsearch ← Index Pattern
  - Distribution in Time Period ← Time Picker
- Make sure to set the correct index pattern and timeframe:



+ Add filter

apachelogs*

Selected fields

? _source

Available fields

- @timestamp
- t _id
- t _index
- # _score
- t _type
- t agent

Aug 26, 2014 @ 01:45:26.041 - Sep 26, 2014 @ 10:23:09.723 –

Auto

Time	_source
Sep 25, 2014 @ 14:00:00.000	geoip.latitude: 23.117 geoip.region_name: Guangdong geoip.country_code3: CN geoip.ip: 183.60.215.50 geoip.continent_code: AS geoip.city_name: Guangzhou geoip.timezone: Asia/Shanghai geoip.region_code: GD geoip.location: { "lon": 113.25, "lat": 23.1167 } geoip.longitude: 113.25

Demo

```
Sep 25, 2014 @ 14:00:00.000   geoip.latitude: 23.117 geoip.region_name: Guangdong geoip.country_code3: CN geoip.ip: 183.60.215.50
                              geoip.continent_code: AS geoip.city_name: Guangzhou geoip.timezone: Asia/Shanghai
                              geoip.region_code: GD geoip.location: { "lon": 113.25, "lat": 23.1167 } geoip.longitude: 113.25
                              geoip.country_name: China geoip.country_code2: CN @timestamp: Sep 25, 2014 @ 14:00:00.000 verb: GET
                              referrer: "-" request: /scripts/netcat-webserver useragent.device: Spider useragent.os: Other

Sep 25, 2014 @ 13:59:29.000   geoip.latitude: 42.886 geoip.region_name: New York geoip.dma_code: 514 geoip.country_code3: US
                              geoip.ip: 198.46.149.143 geoip.continent_code: NA geoip.postal_code: 14202 geoip.city_name: Buffalo
                              geoip.timezone: America/New_York geoip.region_code: NY geoip.location: { "lon": -78.8781, "lat":
                              42.8864 } geoip.longitude: -78.878 geoip.country_name: United States geoip.country_code2: US
```

Instructor Demo

Lesson 2

Review - Discover Interface

Summary

- The discover interface allows you to explore the different aspects of your data
- The most common mistake in the discover interface is not checking the **index pattern** and **time picker**
- The search bar can be used to search all the data inside Elasticsearch
- The document table can be customized to display a table of only selected fields

Quiz

1. What are the first two settings someone should check when using the discover interface?
2. What are the three different boolean operators?
3. Build the query: "Find the messages from **Claire** younger than **30** years old that belong to the category **A**?"

Lesson 2

Lab - Discover Interface

Lesson 3

Aggregations

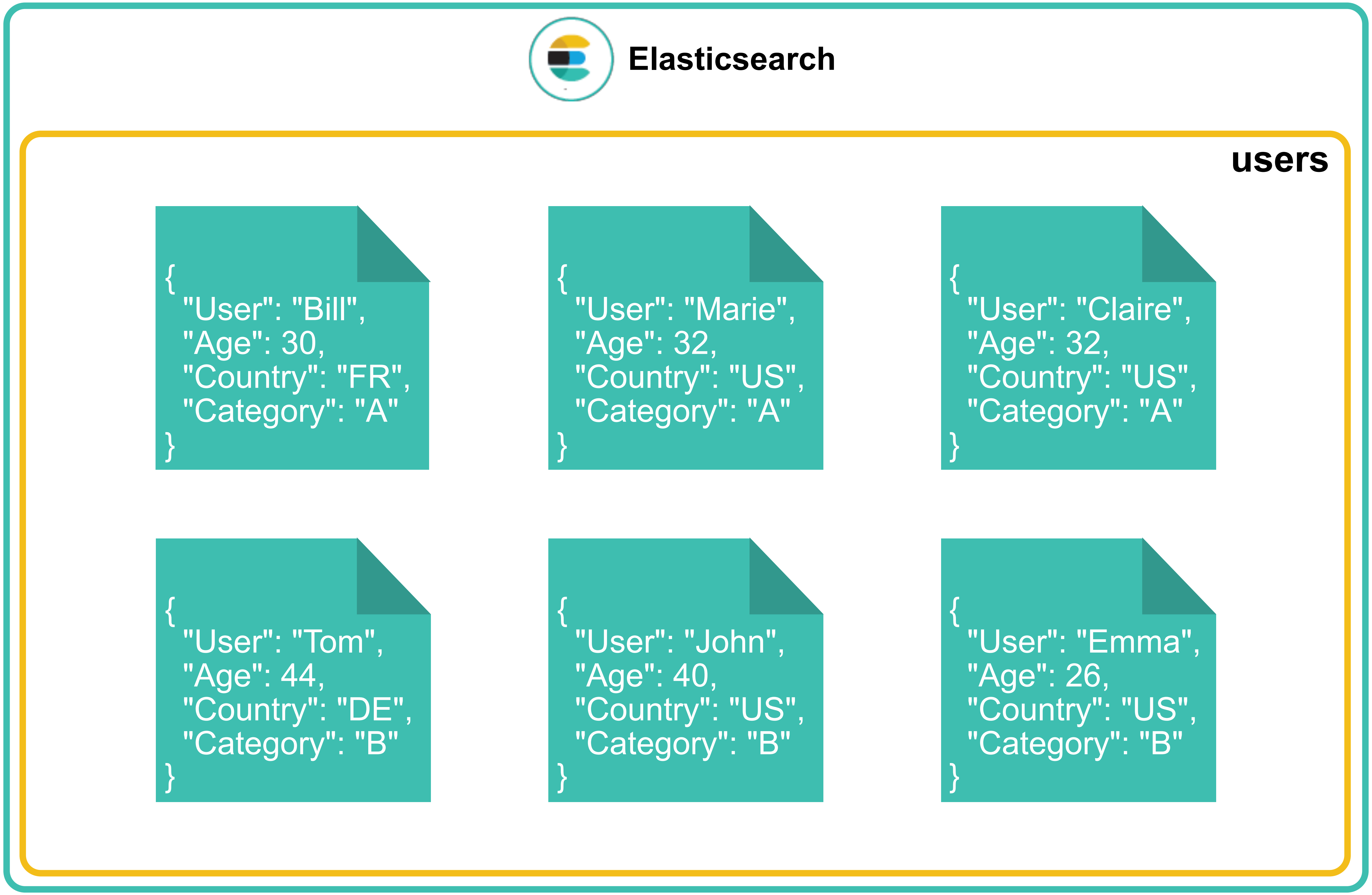
Overview

- Data is often complex and involves many dimensions
- Often, we want summarized insights:
  - slices based on specific attributes
  - calculations based on specific attributes
  - ...
- Spreadsheets might fulfill this using a "**pivot table**"
- In the Elastic Stack we call the equivalent functionality an **aggregation**
- All aggregations are performed at elasticsearch, Kibana just renders the results

A Simple Example: Spreadsheet

id	user	age	country	category
1	Bill	30	FR	A
2	Marie	32	US	A
3	Claire	32	US	A
4	Tom	44	DE	B
5	John	40	US	B
6	Emma	26	US	B

A Simple Example: Elasticsearch



Elasticsearch

users

```
{
  "User": "Bill",
  "Age": 30,
  "Country": "FR",
  "Category": "A"
}
```

```
{
  "User": "Marie",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Claire",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Tom",
  "Age": 44,
  "Country": "DE",
  "Category": "B"
}
```

```
{
  "User": "John",
  "Age": 40,
  "Country": "US",
  "Category": "B"
}
```

```
{
  "User": "Emma",
  "Age": 26,
  "Country": "US",
  "Category": "B"
}
```

Metrics Aggregation

- Metric aggregations
  - Calculates numerical values over a set of documents
  - similar to how **values** are **summarized** in a **pivot table** for a specific column
  - mathematic operation that outputs
    - a single value (eg., **avg**, **sum**, **min**, **max**, **unique count**)
    - or multiple values (eg., **percentiles**, **percentile_ranks**)

A Simple Average Using Pivot Table

id	user	age	country	category
1	Bill	30	FR	A
2	Marie	32	US	A
3	Claire	32	US	A
4	Tom	44	DE	B
5	John	40	US	B
6	Emma	26	US	B

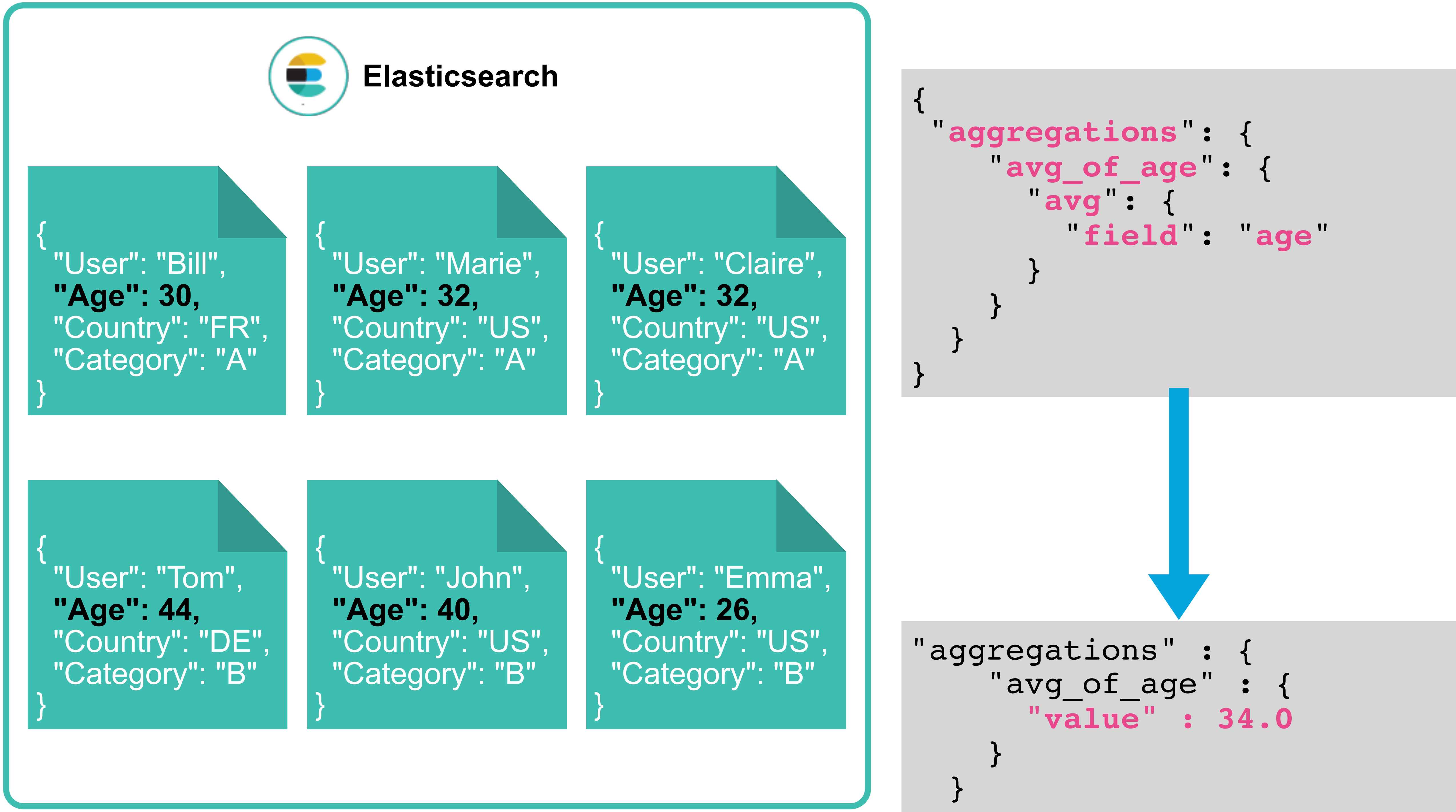
Pivot table definition

Rows	Values
	AVG of age

Pivot table

AVG of age
34

A Simple Average Using Aggregations



Elasticsearch

```
{
  "User": "Bill",
  "Age": 30,
  "Country": "FR",
  "Category": "A"
}
```

```
{
  "User": "Marie",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Claire",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Tom",
  "Age": 44,
  "Country": "DE",
  "Category": "B"
}
```

```
{
  "User": "John",
  "Age": 40,
  "Country": "US",
  "Category": "B"
}
```

```
{
  "User": "Emma",
  "Age": 26,
  "Country": "US",
  "Category": "B"
}
```

```
{
 "aggregations": {
    "avg_of_age": {
      "avg": {
        "field": "age"
      }
    }
  }
}
```

```
"aggregations" : {
    "avg_of_age" : {
      "value" : 34.0
    }
  }
```

Buckets

- Bucket aggregation
  - A way of **slicing** data
  - similar to grouping by values in **rows** or **columns** in a **pivot table**
  - Creates ***buckets***
    - collection of documents that share a common criterion
    - can have one or more metrics associated with it
    - Number of documents (doc count) per bucket is default metric

Simple Bucket Using a Pivot Table

id	user	age	country	category
1	Bill	30	FR	A
2	Marie	32	US	A
3	Claire	32	US	A
4	Tom	44	DE	B
5	John	40	US	B
6	Emma	26	US	B

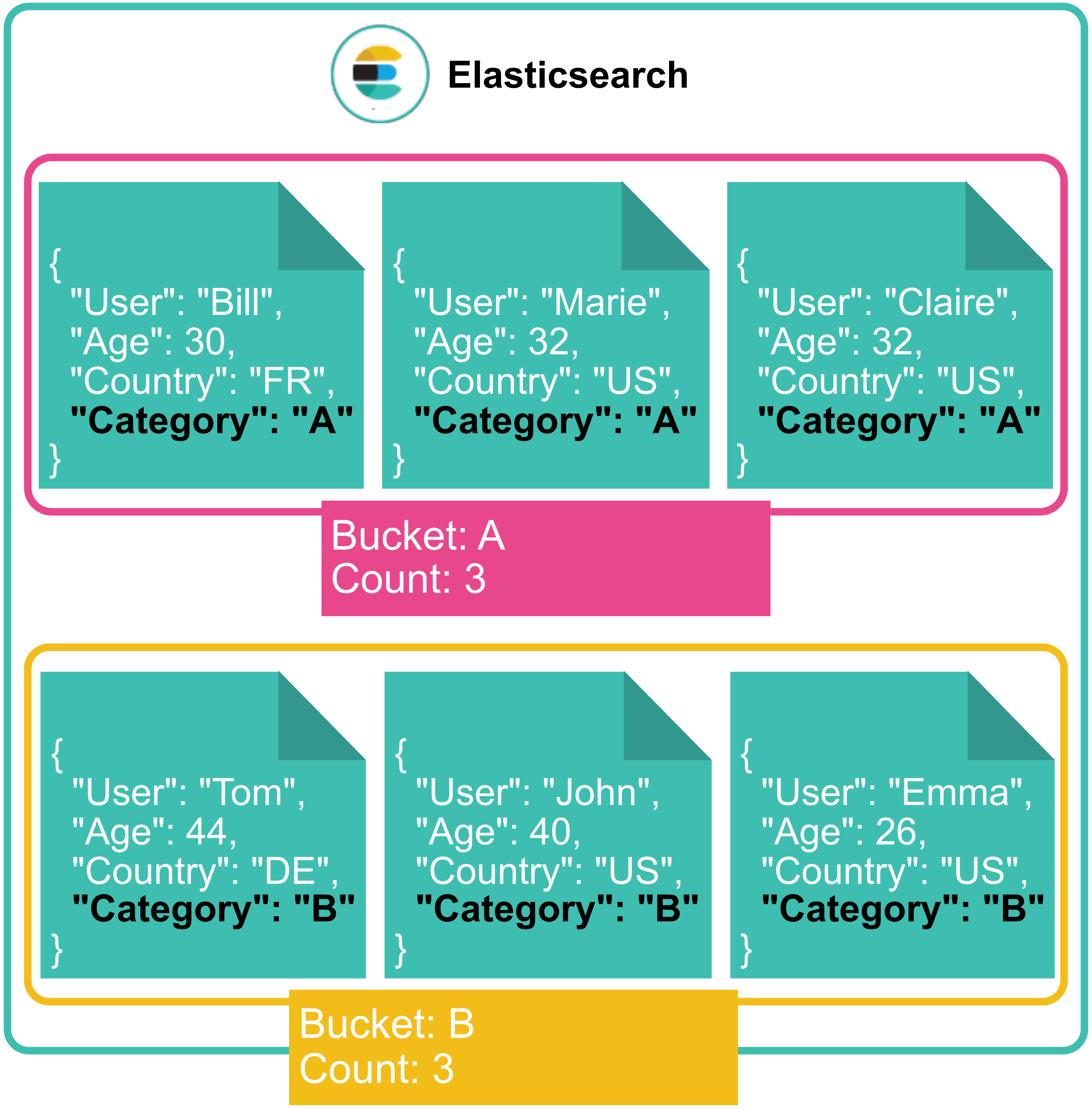
Pivot table definition

Rows	Values
Order ASC by category	COUNT of id

Pivot table

category	COUNT of id
A	3
B	3

Simple Bucket Aggregation



Elasticsearch

```
{
  "User": "Bill",
  "Age": 30,
  "Country": "FR",
  "Category": "A"
}
```

```
{
  "User": "Marie",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Claire",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

Bucket: A
Count: 3

```
{
  "User": "Tom",
  "Age": 44,
  "Country": "DE",
  "Category": "B"
}
```

```
{
  "User": "John",
  "Age": 40,
  "Country": "US",
  "Category": "B"
}
```

```
{
  "User": "Emma",
  "Age": 26,
  "Country": "US",
  "Category": "B"
}
```

Bucket: B
Count: 3

```
{
 "aggregations": {
    "categories": {
      "terms": {
        "field": "category"
      }
    }
  }
}
```

```
"aggregations": {
    "categories": {
    "buckets": [
         {
           "key": "A",
           "doc_count": 3
         },
         {
           "key": "B",
           "doc_count": 3
         }
       ]
    }
  }
```

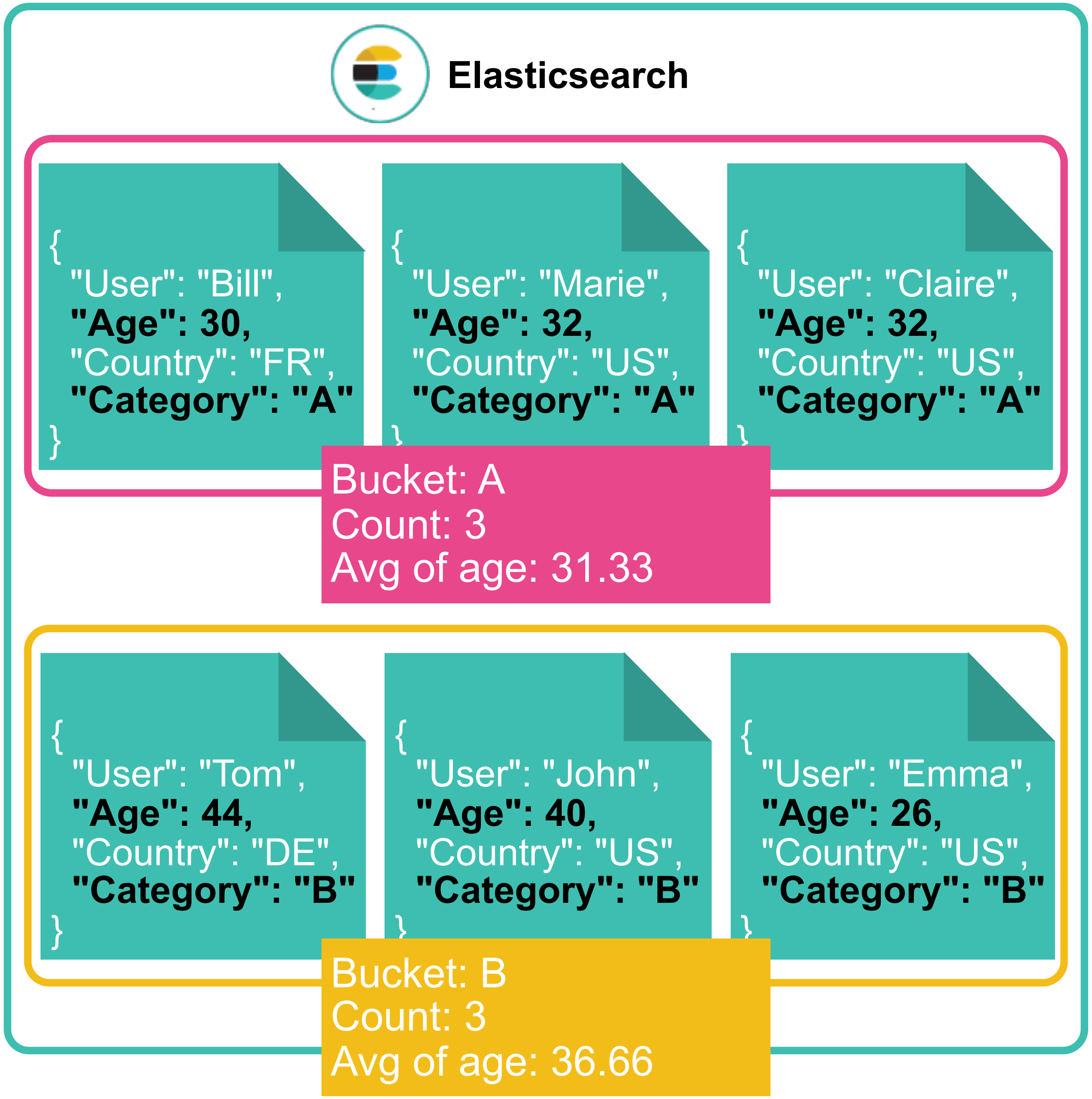
Adding Metrics

id	user	age	country	category
1	Bill	30	FR	A
2	Marie	32	US	A
3	Claire	32	US	A
4	Tom	44	DE	B
5	John	40	US	B
6	Emma	26	US	B

Rows	Values
Order ASC by category	COUNT of age
	AVG of age

category	COUNT of age	AVG of age
A	3	31.33
B	3	36.66

Adding Metrics



Elasticsearch

```
{
  "User": "Bill",
  "Age": 30,
  "Country": "FR",
  "Category": "A"
}
```

```
{
  "User": "Marie",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Claire",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

Bucket: A
Count: 3
Avg of age: 31.33

```
{
  "User": "Tom",
  "Age": 44,
  "Country": "DE",
  "Category": "B"
}
```

```
{
  "User": "John",
  "Age": 40,
  "Country": "US",
  "Category": "B"
}
```

```
{
  "User": "Emma",
  "Age": 26,
  "Country": "US",
  "Category": "B"
}
```

Bucket: B
Count: 3
Avg of age: 36.66

```
"aggregations": {
  "categories": {
    "terms": {
      "field": "category"
    },
    "aggregations": {
      "avg_age_per_category": {
        "avg": {
          "field": "age"
        }
} } } }
```

```
"aggregations": {
  "categories": {
    "buckets": [
      {
        "key": "A",
        "doc_count": 3,
        "avg_age_per_category": {
          "value": 31.33
        }
      },
      {
        "key": "B",
        "doc_count": 3,
        "avg_age_per_category": {
          "value": 36.66
        }
} ] } }
```

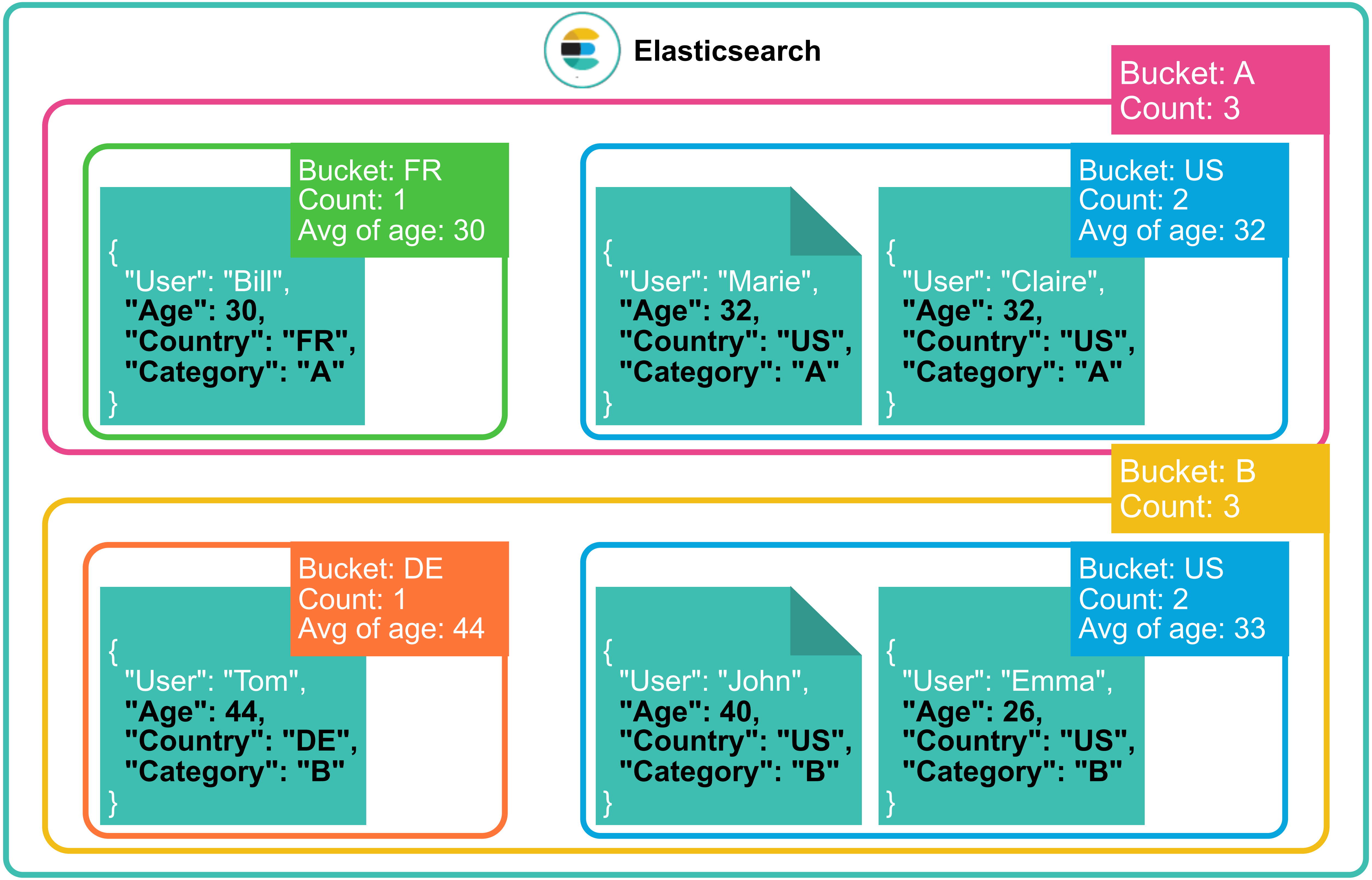
Nesting Rows/Columns in a Pivot Table

id	user	age	country	category
1	Bill	30	FR	A
2	Marie	32	US	A
3	Claire	32	US	A
4	Tom	44	DE	B
5	John	40	US	B
6	Emma	26	US	B

Rows	Values
Order ASC by category	COUNT of age
Order ASC by country	AVG of age

category	country	COUNT of age	AVG of age
A	FR	1	30
	US	2	32
B	DE	1	44
	US	2	33

Adding Sub-Bucket Aggregation



Elasticsearch

Bucket: A
Count: 3

Bucket: FR
Count: 1
Avg of age: 30

```
{
  "User": "Bill",
  "Age": 30,
  "Country": "FR",
  "Category": "A"
}
```

Bucket: US
Count: 2
Avg of age: 32

```
{
  "User": "Marie",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

```
{
  "User": "Claire",
  "Age": 32,
  "Country": "US",
  "Category": "A"
}
```

Bucket: B
Count: 3

Bucket: DE
Count: 1
Avg of age: 44

```
{
  "User": "Tom",
  "Age": 44,
  "Country": "DE",
  "Category": "B"
}
```

Bucket: US
Count: 2
Avg of age: 33

```
{
  "User": "John",
  "Age": 40,
  "Country": "US",
  "Category": "B"
}
```

```
{
  "User": "Emma",
  "Age": 26,
  "Country": "US",
  "Category": "B"
}
```

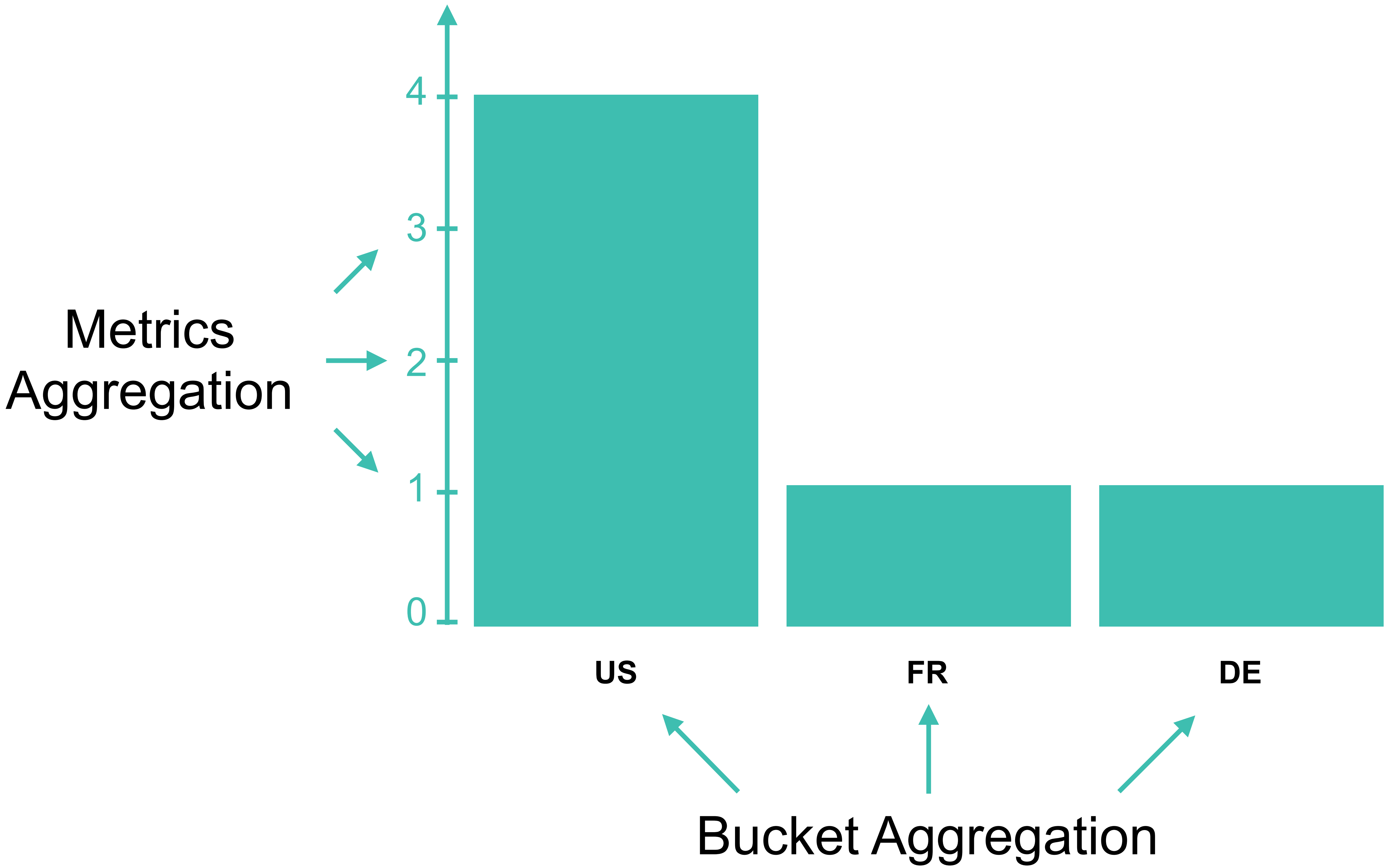
Metrics Aggregation

Metrics Aggregation

6

Count of Documents

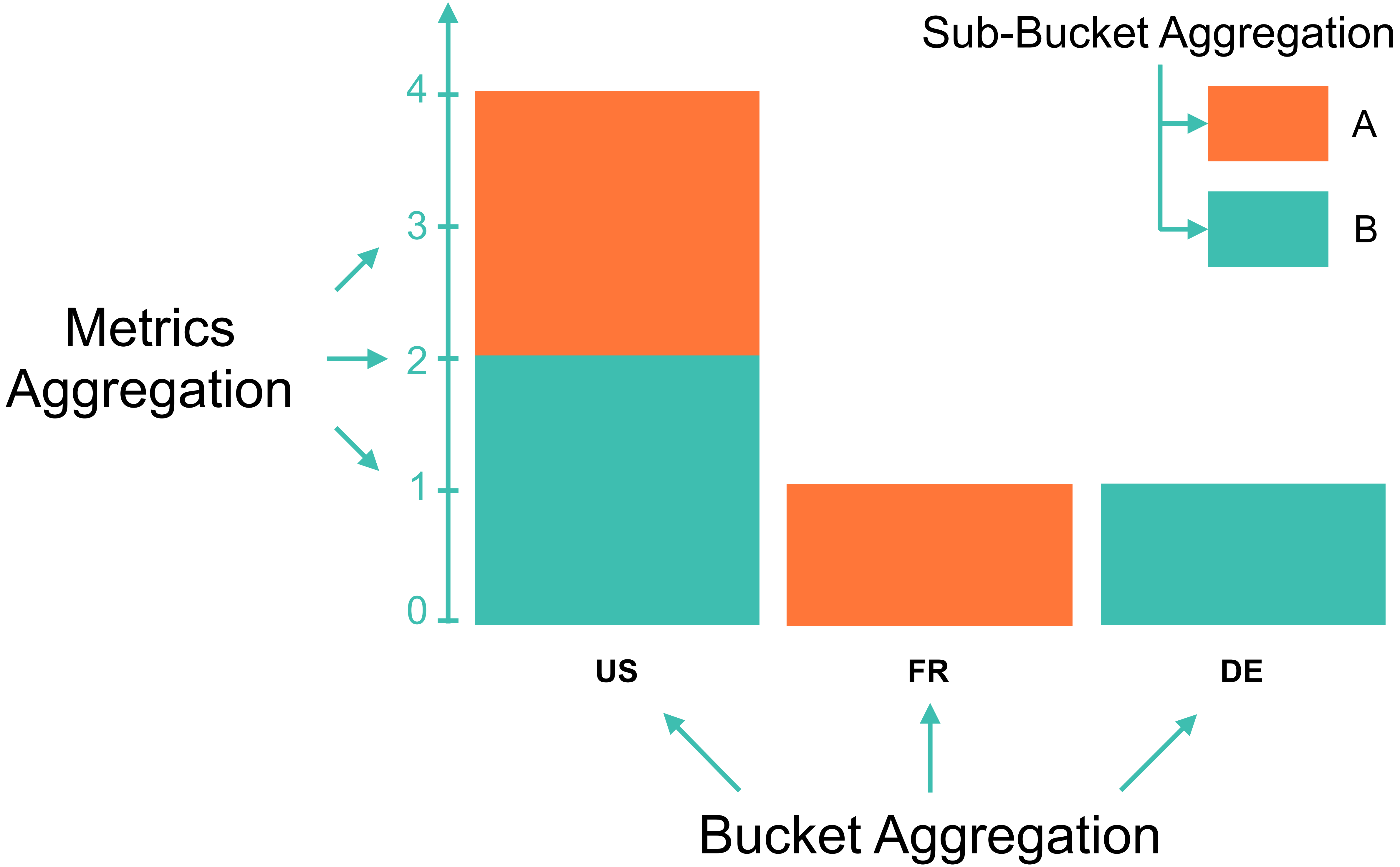
Bucket Aggregation



Metrics Aggregation

Bucket Aggregation

Sub-bucket Aggregation



Lesson 3

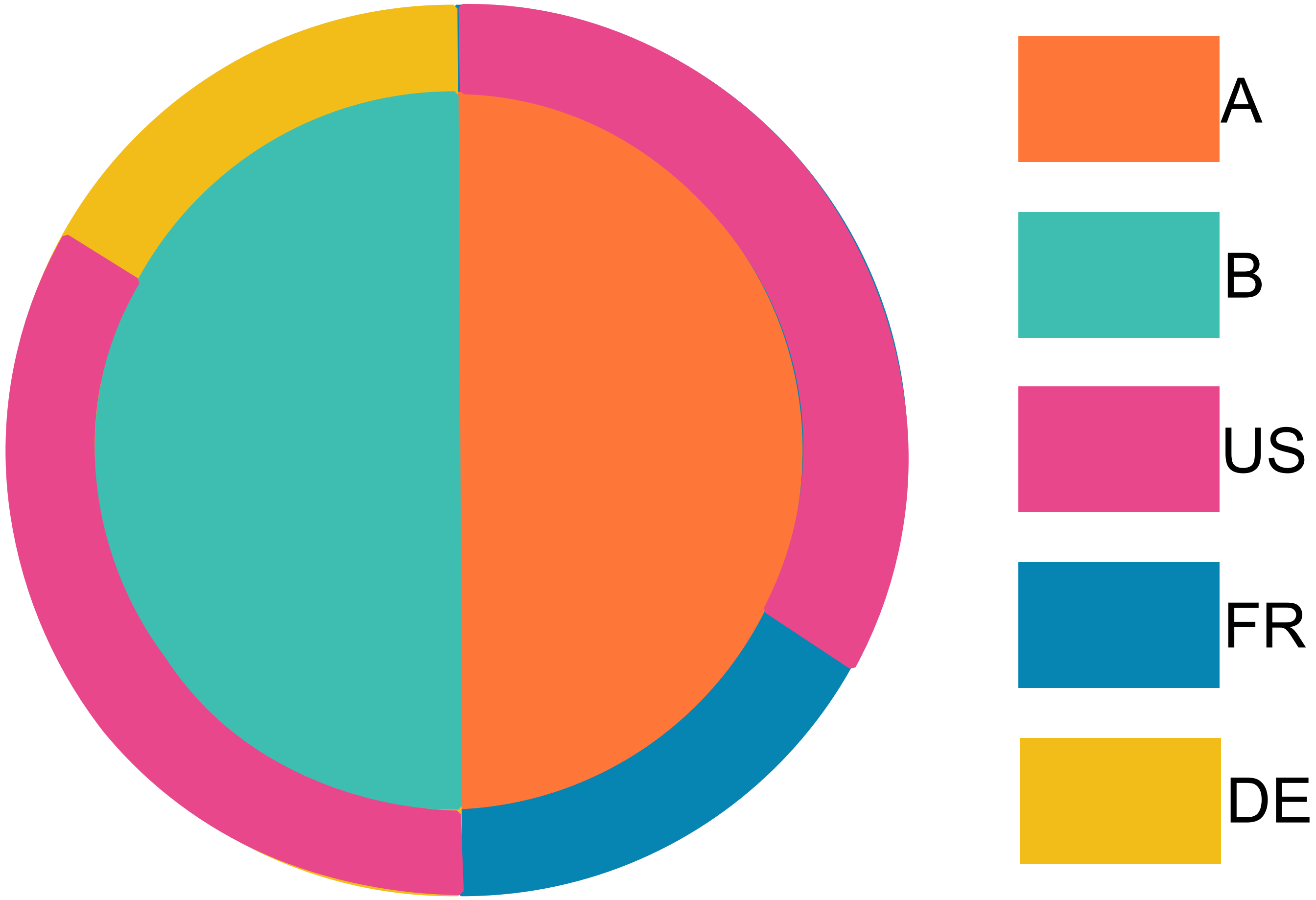
Review - Aggregations

Summary

- Kibana renders visualizations using the results of Elasticsearch aggregations
- There are two main types of aggregations:
  - metric
  - bucket
- Metric aggregations are used to compute numeric values
- Bucket aggregations are used to group data together

Quiz

1. What are the two main types of aggregations?
2. **True or False:** Aggregations are used by Kibana to render visualizations.
3. Explain which aggregations are used to build the following visualization.



Lesson 3

Lab - Aggregations