Elasticsearch

IN4120

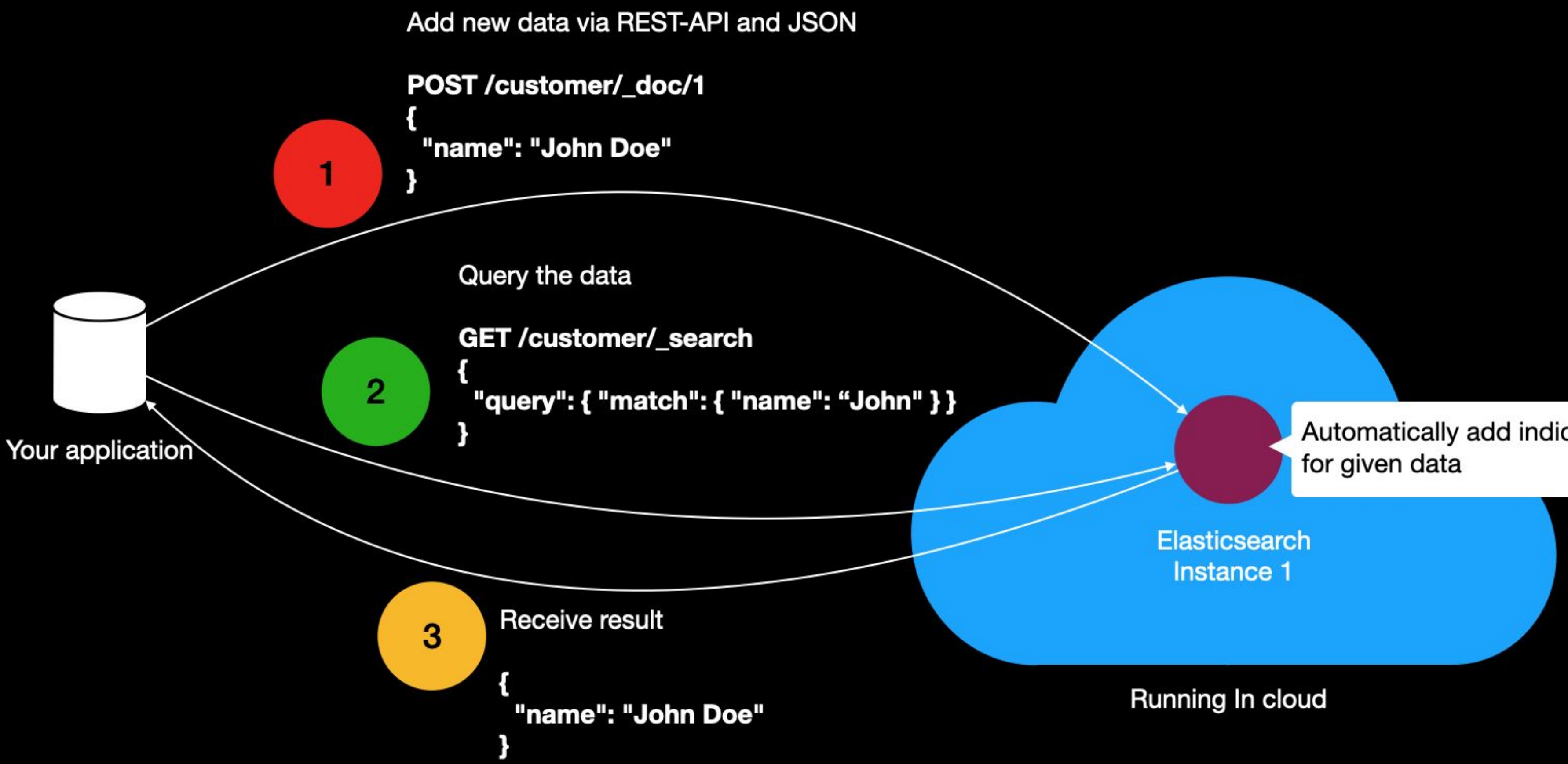
Hugo Nørholm and Jan Grünwaldt, November 16th

What is Elasticsearch - as simple as possible

- Elasticsearch is a search engine developed by Elastic
- You can download it and run it locally or run it in the cloud
- It has many different use cases, e.g. adding search to your application, or log monitoring and analysis
- The magic of it is: you only add the data, Elasticsearch does all the indexing and scaling by itself
- You can then very quickly search the data, even at large scale

What is Elasticsearch - as simple as possible

- It is built on top of Apache Lucene
- Lucene is a Java library that provides functions to create your own inverted index and search engine
- Basically, imagine you took all your assignment code, get rid of most simplifications (e.g. keeping everything in memory) and turned it into a library
- Then, what is Elasticsearch?
- Elasticsearch is an additional layer on top of it:
    - JSON-based REST-API
    - Distributed architecture: multiple separate Lucene instances
    - Also provides montoring and managing of different instances, and much more
    - You can use Lucene, without knowing its syntax or caring about managing instances



1

Add new data via REST-API and JSON

```
POST /customer/_doc/1
{
  "name": "John Doe"
}
```

2

Query the data

```
GET /customer/_search
{
  "query": { "match": { "name": “John" } }
}
```

3

Receive result

```
{
  "name": "John Doe"
}
```

Your application

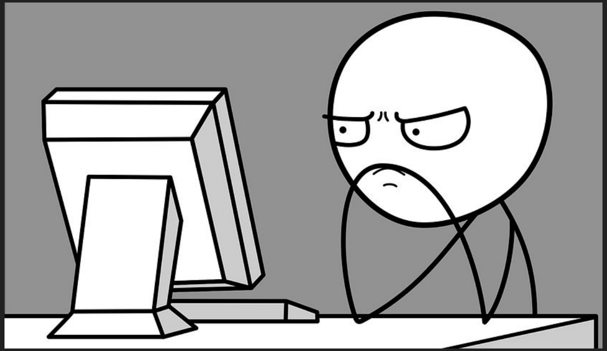
Automatically add indic
for given data

Elasticsearch
Instance 1

Running In cloud

Another example: log monitoring with ELK stack

- Let’s say you developed your fancy new web app and want to serve it to the world
- After successfully launching in the morning, you invite your team to celebrate
- Unfortunately, you are unaware that your server crashed, and the users are unhappy



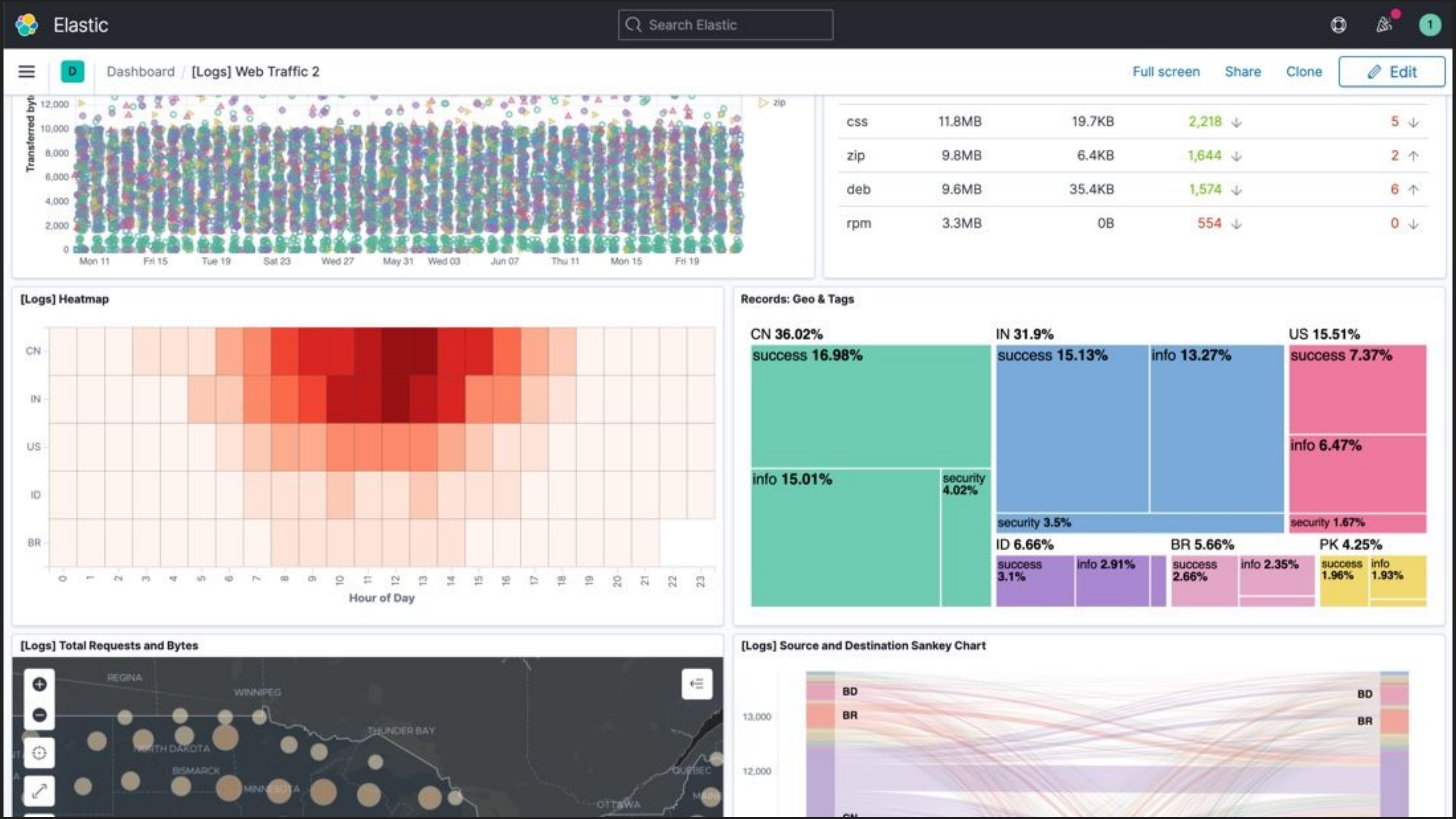
Source: https://pixabay.com/vectors/computer-internet-unhappy-user-1295358/

Another example: log monitoring with ELK stack

- You did not notice in time and have no idea what really happened
- So you decide: you need to monitor your server logs

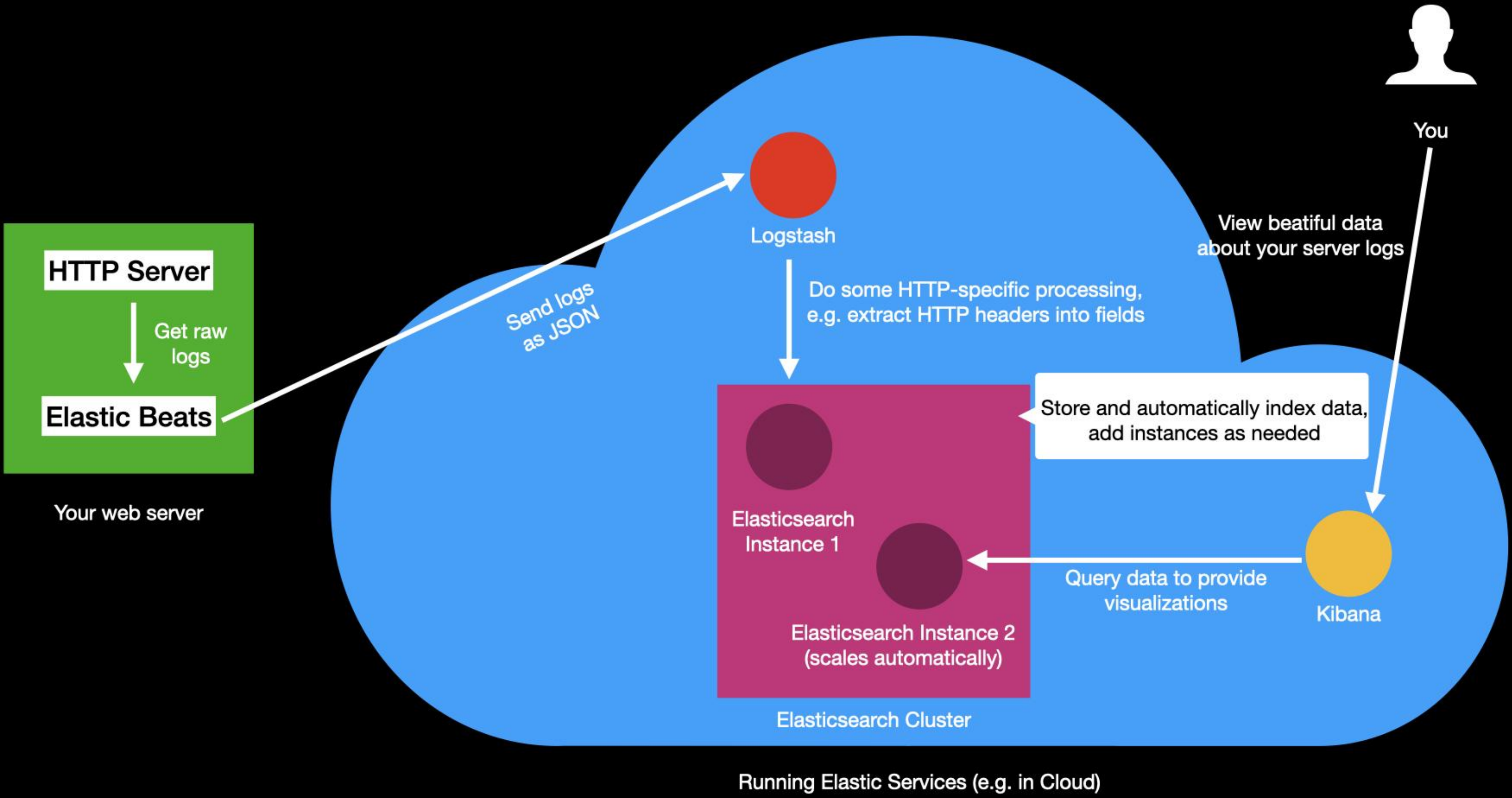
→ And here is where the “Elastic stack” comes in handy

Elasticsearch is one component of this stack



css	11.8MB	19.7KB	2,218 ↓	5 ↓
zip	9.8MB	6.4KB	1,644 ↓	2 ↑
deb	9.6MB	35.4KB	1,574 ↓	6 ↑
rpm	3.3MB	0B	554 ↓	0 ↓

Apache Kibana, Source: https://www.elastic.co/kibana/



You

HTTP Server

Get raw logs

Elastic Beats

Your web server

Send logs as JSON

Logstash

Do some HTTP-specific processing, e.g. extract HTTP headers into fields

View beatiful data about your server logs

Store and automatically index data, add instances as needed

Elasticsearch Instance 1

Elasticsearch Instance 2 (scales automatically)

Query data to provide visualizations

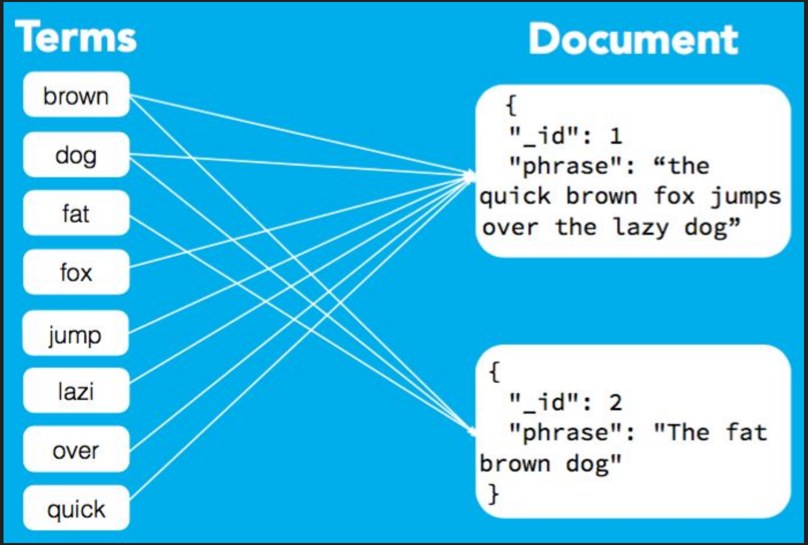
Kibana

Elasticsearch Cluster

Running Elastic Services (e.g. in Cloud)

How does it work - logical concepts

- Documents
  - Similar to a row entry in a relational database
  - Any structured data encoded in JSON
- Indices
  - Highest level entity
  - Similar to a database in a relational database
  - For example customers, products and orders
- Inverted index
  - Does not store whole documents
  - Splits into search terms
  - The foundation for quick efficient search



```
{
 "_id": 1
 "phrase": “the
quick brown fox jumps
over the lazy dog”
```

```
{
 "_id": 2
 "phrase": "The fat
brown dog"
}
```

Source: https://dzone.com/articles/elasticsearch-101

How does it work - backend structure

- Cluster - a group of nodes that are connected together which allow for distribution of tasks, indexing and searching across the different nodes
- Node - a single server that is part of a cluster, can be configured in several ways
    - Master node - responsible for creating/deleting an index and adding/removing nodes
    - Data node - stores the data and handles search and aggregation requests
    - Client node - forwards server requests to the master node and data-related requests to data nodes.
- Shards subdivided index that can be spread across different nodes, each shard functions as its own index
    - Replica shards - Duplicated shards that provide redundant copies that protect against hardware failures.

What to learn from this?

- While the course is focused on the theoretic ideas that made building search engines possible, today we can learn features that matter in practice:
  - Adding data to the system from heterogeneous sources
  - Integrate with existing systems (via REST-API)
  - Generate additional value out of the data: e.g. visualize it, send alerts based on data, use ML on the data
  - Run the search engine as you like: e.g. self-hosted, as-a Service, or self-managed cloud
  - Scale automatically from very small to very large datasets

Sources

- https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html (accessed Nov 2, 2022)
- https://www.bmc.com/blogs/elasticsearch-logs-beats-logstash/ (accessed Nov 2, 2022)
- https://www.knowi.com/blog/what-is-elastic-search/ (accessed Nov 2, 2022)
- https://www.alibabacloud.com/blog/what-is-elasticsearch-and-how-does-elasticsearch-work_597235 (accessed Nov 13, 2022)
- https://www.elastic.co/beats/ (accessed Nov 2, 2022)
- https://www.elastic.co/logstash/ (accessed Nov 2, 2022)
- https://www.elastic.co/kibana/ (accessed Nov 2, 2022)