Elasticsearch

A central component of the Elastic Stack

Huicong Jiang Yaqing Cao Yuri Liao

Group 8

elastic

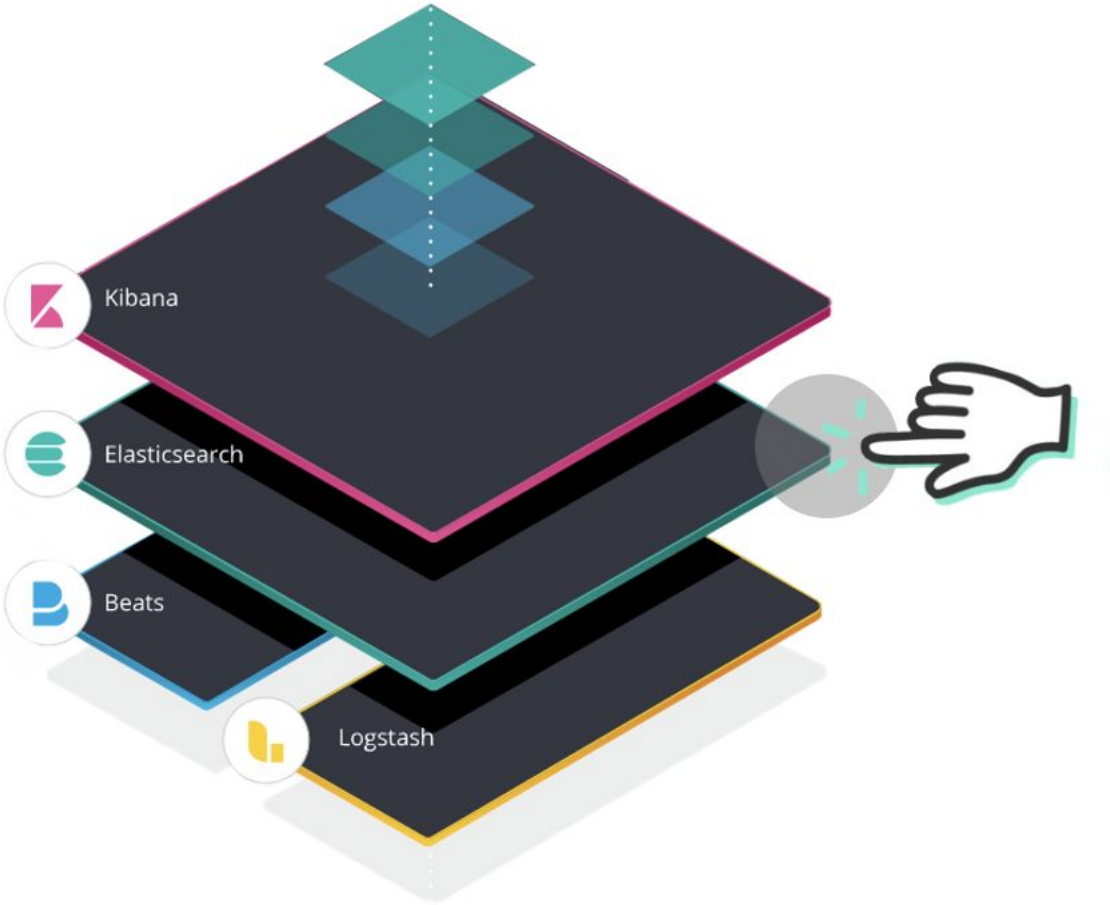


Figure 1: Retrieved from https://www.elastic.co/what-is/elk-stack

fast and scalable

kind of like Google

a big data solution

a search engine

an index.

an “analytics database”

What is Elasticsearch?

distributed

open-source search

analytics engine

Apache Lucene

Java

Elasticsearch vs MySQL comparison

Database → index

Table → Type

Column → key/value pair(Filed)

Row → document

Inquiry → GET/POST/DELETE/PUT

History of Elasticsearch

2000

Recipe app - Compass

Shay Banon,
Steven Schuurman,
Uri Boness,
Simon Willnauer

2010

Released first version

2012

Provided commercial services

2015

Versioning bug
Elasticsearch → Elastic
Connected directly with AWS Elastic cloud

2018

Opened commercial code
Listed on the New York Stock Exchange

User cases

More details in https://www.elastic.co/customers/?elektra=home&storm=logo-bar

Advantages

Full-text search

Compatibility

Fast performance

Schema-free

Open source

Scalability

Disadvantages

Steep learning curve

Handling format constraint

Memory requirement

Storage capacity limit

Fine-tuned / High maintenance

Cost

01. Open source

free to download; rent servers; effort on management

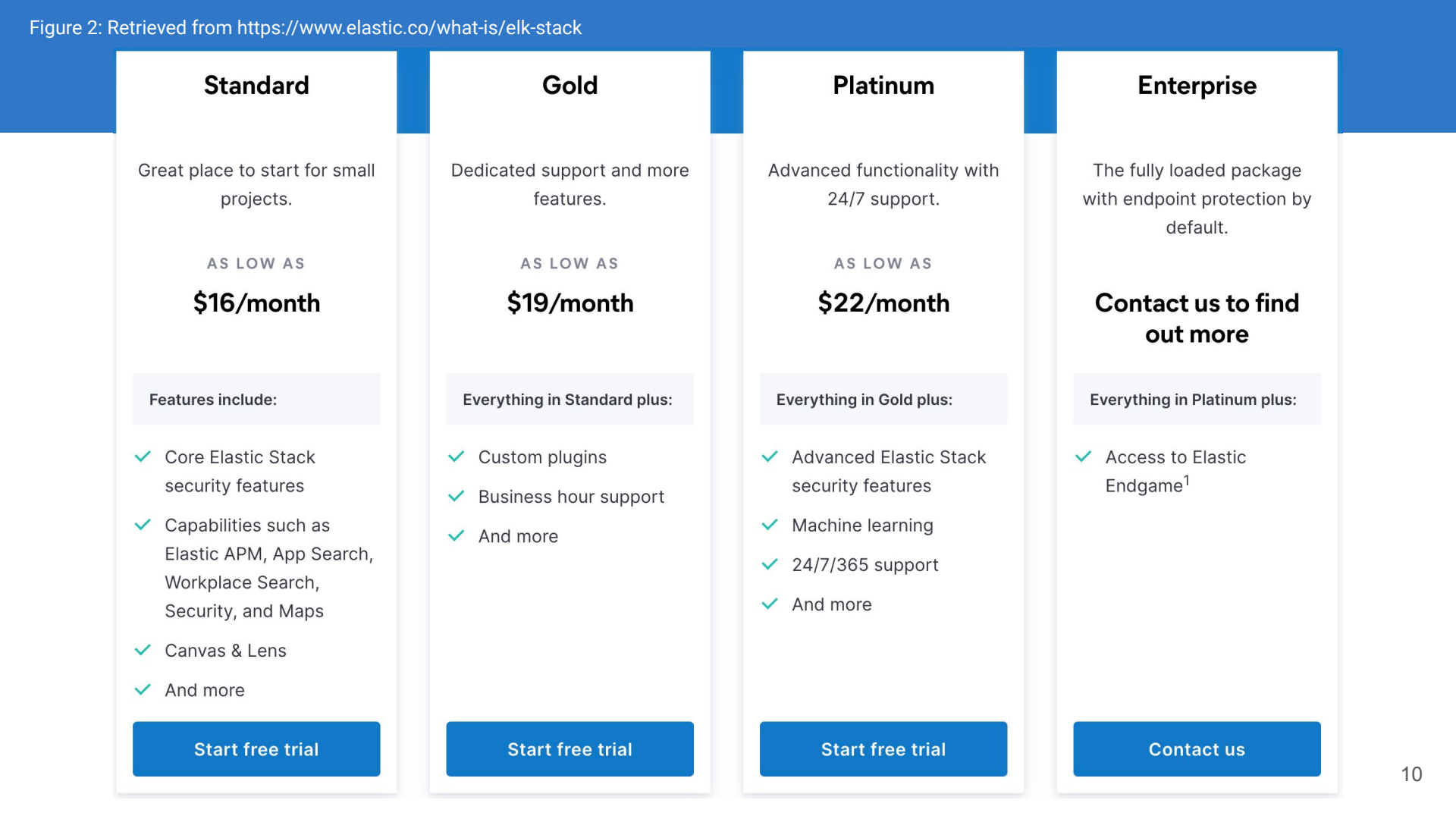
02. elastic.co

basic month fee: $16/month - $22/month - more; additional charges

03. Amazon Elasticsearch Service

different instances and services; expensive

Figure 2: Retrieved from https://www.elastic.co/what-is/elk-stack



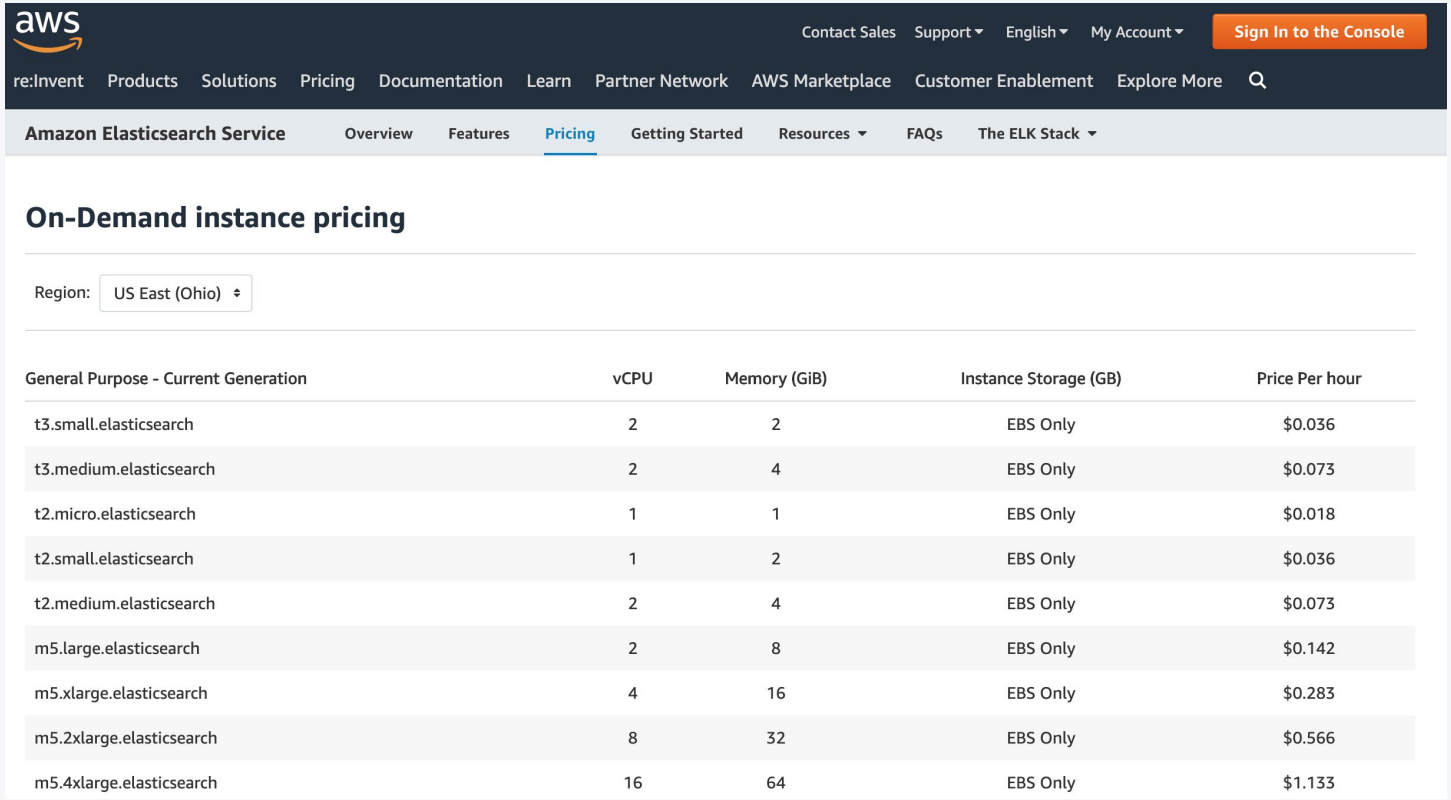
How is pricing for Elasticsearch Service on Elastic Cloud calculated?

Pricing for Elasticsearch Service is primarily based on the instance size of each component of the Elastic Stack, such as Elasticsearch nodes, Kibana nodes, APM server, etc. Each component that is running is charged per GB of RAM/hour. You can estimate your costs using our pricing calculator. This rate is based on the following dimensions:

- Cloud provider
- Cloud region
- Elasticsearch nodes
  - Data nodes
    - High I/O
    - High storage
    - High CPU

Free tier

You can get started for free on Amazon Elasticsearch Service with the AWS Free Tier. For customers in the AWS Free Tier, Amazon Elasticsearch Service provides free usage of up to 750 hours per month of a t2.small.elasticsearch or t3.small.elasticsearch instance and 10GB per month of optional EBS storage (Magnetic or General Purpose). If you exceed the free tier usage limits, you will be charged the Amazon Elasticsearch Service rates for the additional resources you use. See offer terms for more details.



On-Demand instance pricing

Region: US East (Ohio)

General Purpose - Current Generation	vCPU	Memory (GiB)	Instance Storage (GB)	Price Per hour
t3.small.elasticsearch	2	2	EBS Only	$0.036
t3.medium.elasticsearch	2	4	EBS Only	$0.073
t2.micro.elasticsearch	1	1	EBS Only	$0.018
t2.small.elasticsearch	1	2	EBS Only	$0.036
t2.medium.elasticsearch	2	4	EBS Only	$0.073
m5.large.elasticsearch	2	8	EBS Only	$0.142
m5.xlarge.elasticsearch	4	16	EBS Only	$0.283
m5.2xlarge.elasticsearch	8	32	EBS Only	$0.566
m5.4xlarge.elasticsearch	16	64	EBS Only	$1.133

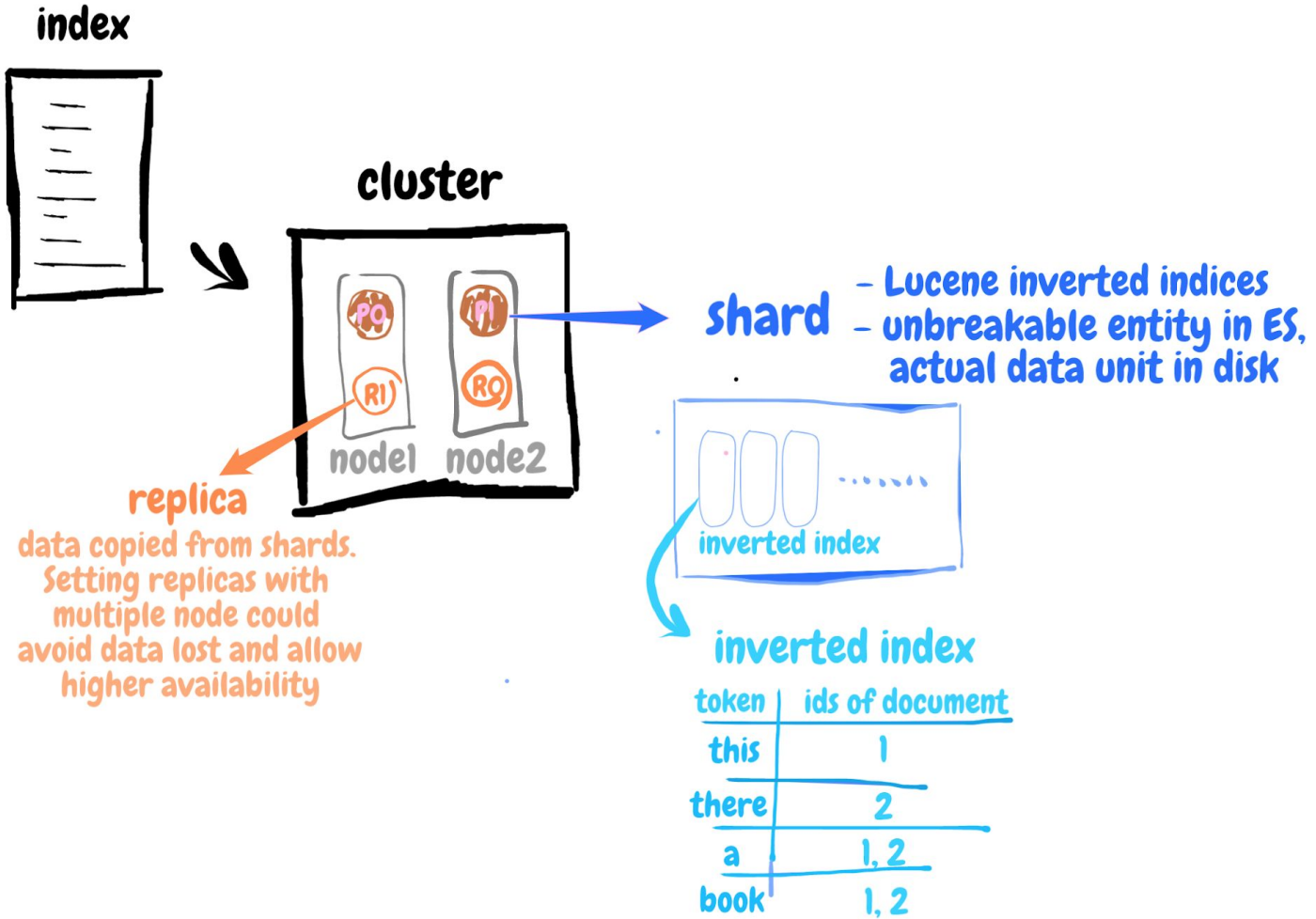
On-Demand instance pricing

Region: US East (Ohio)

General Purpose - Current Generation	vCPU	Memory (GiB)	Instance Storage (GB)	Price Per hour
t3.small.elasticsearch	2	2	EBS Only	$0.036
t3.medium.elasticsearch	2	4	EBS Only	$0.073
t2.micro.elasticsearch	1	1	EBS Only	$0.018
t2.small.elasticsearch	1	2	EBS Only	$0.036
t2.medium.elasticsearch	2	4	EBS Only	$0.073
m5.large.elasticsearch	2	8	EBS Only	$0.142
m5.xlarge.elasticsearch	4	16	EBS Only	$0.283
m5.2xlarge.elasticsearch	8	32	EBS Only	$0.566
m5.4xlarge.elasticsearch	16	64	EBS Only	$1.133

Memory Optimized - Current Generation	vCPU	Memory (GiB)	Instance Storage (GB)	Price Per hour
r5.large.elasticsearch	2	16	EBS Only	$0.186
r5.xlarge.elasticsearch	4	32	EBS Only	$0.372
r5.2xlarge.elasticsearch	8	64	EBS Only	$0.743
r5.4xlarge.elasticsearch	16	128	EBS Only	$1.487
r5.12xlarge.elasticsearch	48	384	EBS Only	$4.46
r4.large.elasticsearch	2	15.25	EBS Only	$0.196
r4.xlarge.elasticsearch	4	30.5	EBS Only	$0.392
r4.2xlarge.elasticsearch	8	61	EBS Only	$0.785
r4.4xlarge.elasticsearch	16	122	EBS Only	$1.569

Storage Optimized - Current Generation	vCPU	Memory (GiB)	Instance Storage (GB)	Price Per hour
i3.large.elasticsearch	2	15.25	1 x 475 NVMe SSD	$0.25
i3.xlarge.elasticsearch	4	30.5	1 x 950 NVMe SSD	$0.499
i3.2xlarge.elasticsearch	8	61	1 x 1900 NVMe SSD	$0.998
i3.4xlarge.elasticsearch	16	122	2 x 1900 NVMe SSD	$1.997
i3.8xlarge.elasticsearch	32	244	4 x 1900 NVMe SSD	$3.994
i3.16xlarge.elasticsearch	64	488	8 x 1900 NVMe SSD	$7.987

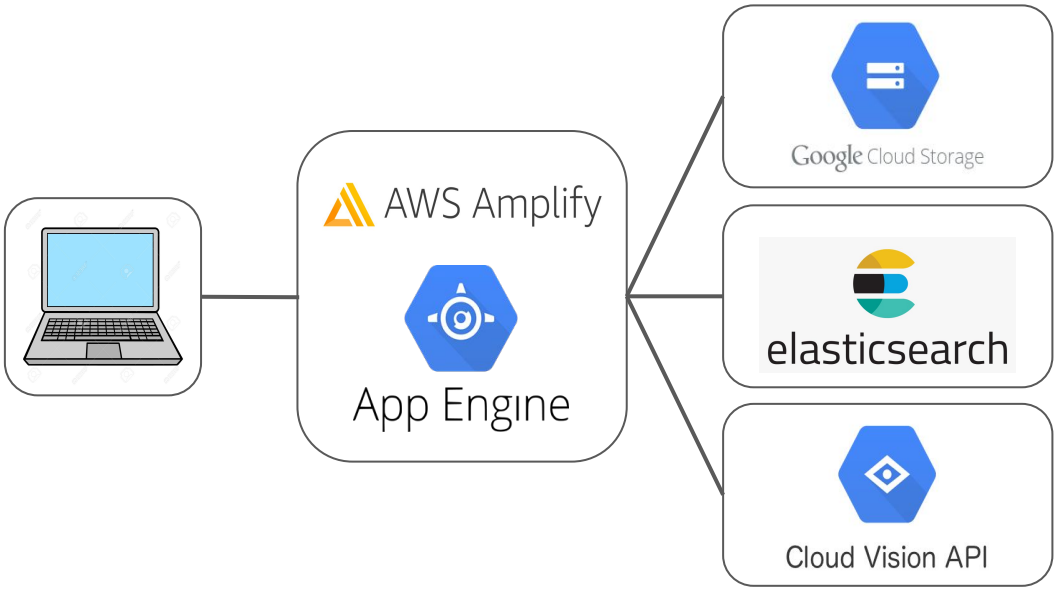

index
cluster
P0
P1
R1
R0
node1
node2
shard
- Lucene inverted indices
- unbreakable entity in ES, actual data unit in disk
inverted index
replica
data copied from shards. Setting replicas with multiple node could avoid data lost and allow higher availability
inverted index
token | ids of document
this | 1
there | 2
a | 1, 2
book | 1, 2

Conclusion

Next: DEMO

Around - A web application

Social network: allow people to share images/videos based on their **geo-location**.



Save objects
Object URL

Index: post → location
Index: user → name

URL : https://test.dm6jj2msx4h8b.amplifyapp.com

Around - A web application

Run Elasticsearch

1. Launch a VM and install JRE
2. Install Elasticsearch
3. Set network and start it
4. Build two index: user, post

```
func main() {
    client, err := elastic.NewClient(elastic.SetURL(ES_URL))
    if err != nil {
        panic(err)
    }

    exists, err := client.IndexExists(POST_INDEX).Do(context.Background())
    if err != nil {
        panic(err)
    }
    if !exists {
        mapping := `{
            "mappings": {
                "properties": {
                    "user":     { "type": "keyword", "index": false },
                    "message":  { "type": "keyword", "index": false },
                    "location": { "type": "geo_point" },
                    "url":      { "type": "keyword", "index": false },
                    "type":     { "type": "keyword", "index": false },
                    "face":     { "type": "float" }
                }
            }
        }`
        _, err = client.CreateIndex(POST_INDEX).Body(mapping).Do(context.Background())
        if err != nil {
            panic(err)
        }
    }
```

URL : https://test.dm6jj2msx4h8b.amplifyapp.com

Around - A web application

RESTful search engine

1. GET http://35.192.145.182:9200/post/_search?q=*:*pretty=true&q=*:*&size=30
2. DELETE http://35.192.145.182:9200/post/_doc/26671e22-5c44-49cc-8e2a-be663a6999ed
3. PUT http://35.192.145.182:9200/post/_doc/1


```
{
        "user": "huicong",
        "message": "Best professor ever",
        "location": {
                "lat": 47.620729,
                "lon": -122.348794
        },
        "url": "https://fakeurl",
        "type": "image",
        "face": 1.0
}
```


URL : https://test.dm6jj2msx4h8b.amplifyapp.com

Around - A web application

1. Google cloud storage

Free

2. Google compute engine

N1-standard-1: 1 cpu, 3.75 Gb memory, $0.04749975/Hour

3. Google cloud storage

Standard storage: Iowa (us-central1) $0.020/Gb/Month

URL : https://test.dm6jj2msx4h8b.amplifyapp.com

Thank you for watching!

Q & A

URL : https://test.dm6jj2msx4h8b.amplifyapp.com