Going Elastic

Philipp Krenn @xeraa

elastic

Infrastructure | Developer Advocate

philipp@elastic.co

Who is using:

- Elasticsearch
- Logstash and Kibana
- Beats

Agenda:

- Overview
- Hands-on
- Questions

While we're getting started

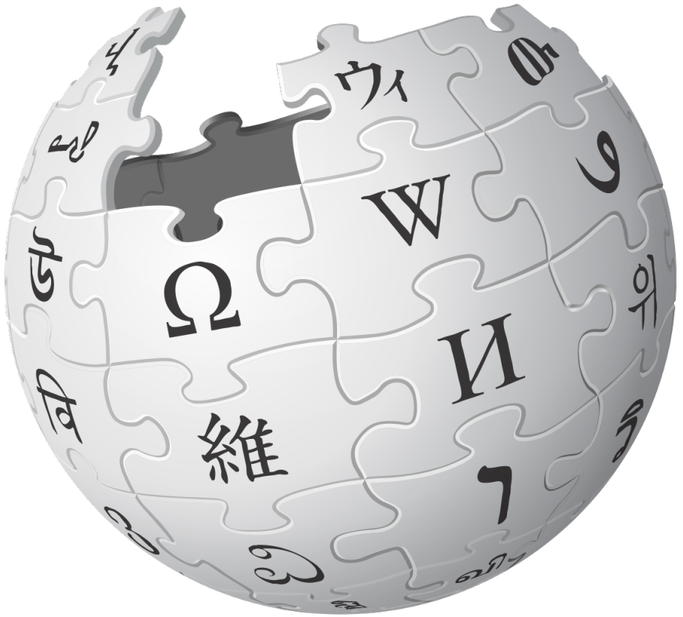
USB Stick

Overview

History

elasticsearch.

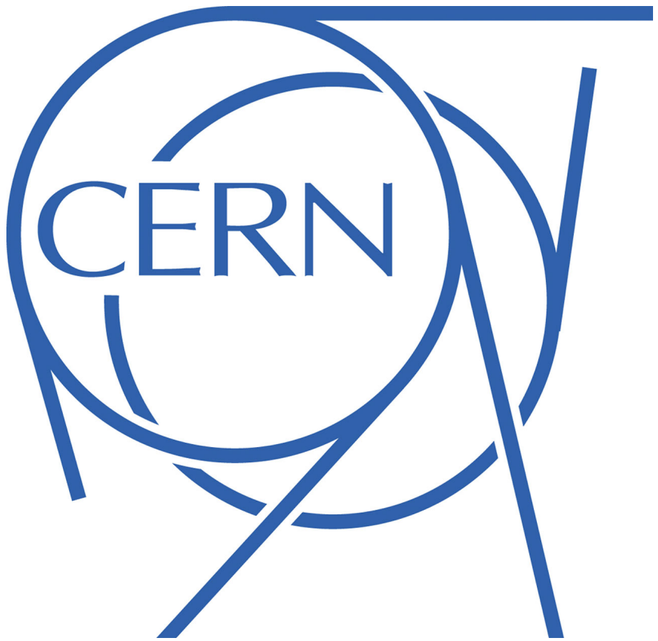
You know, for search



logstash

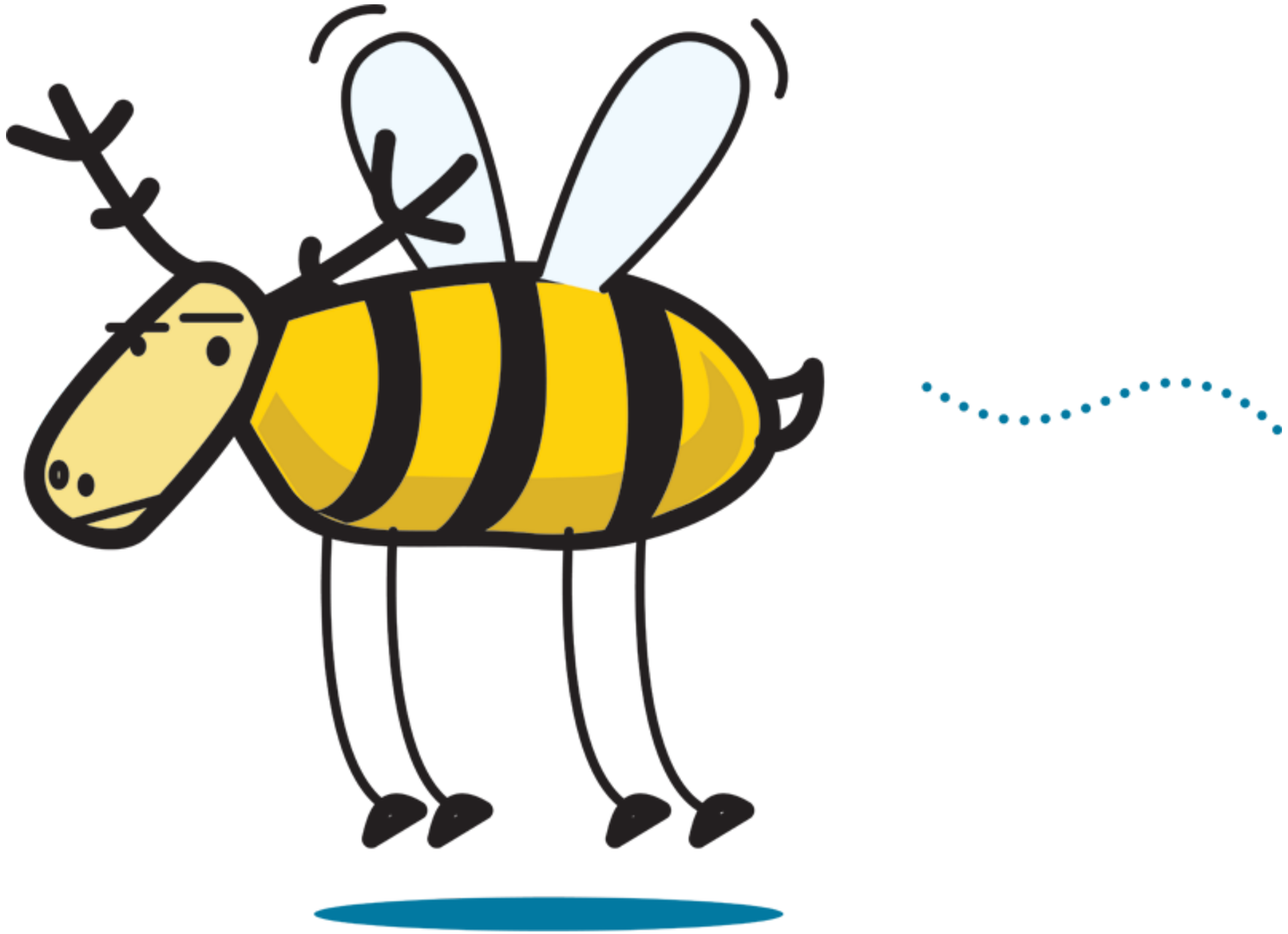
ELK Stack

elastic


CERN


Goldman
Sachs


salesforce

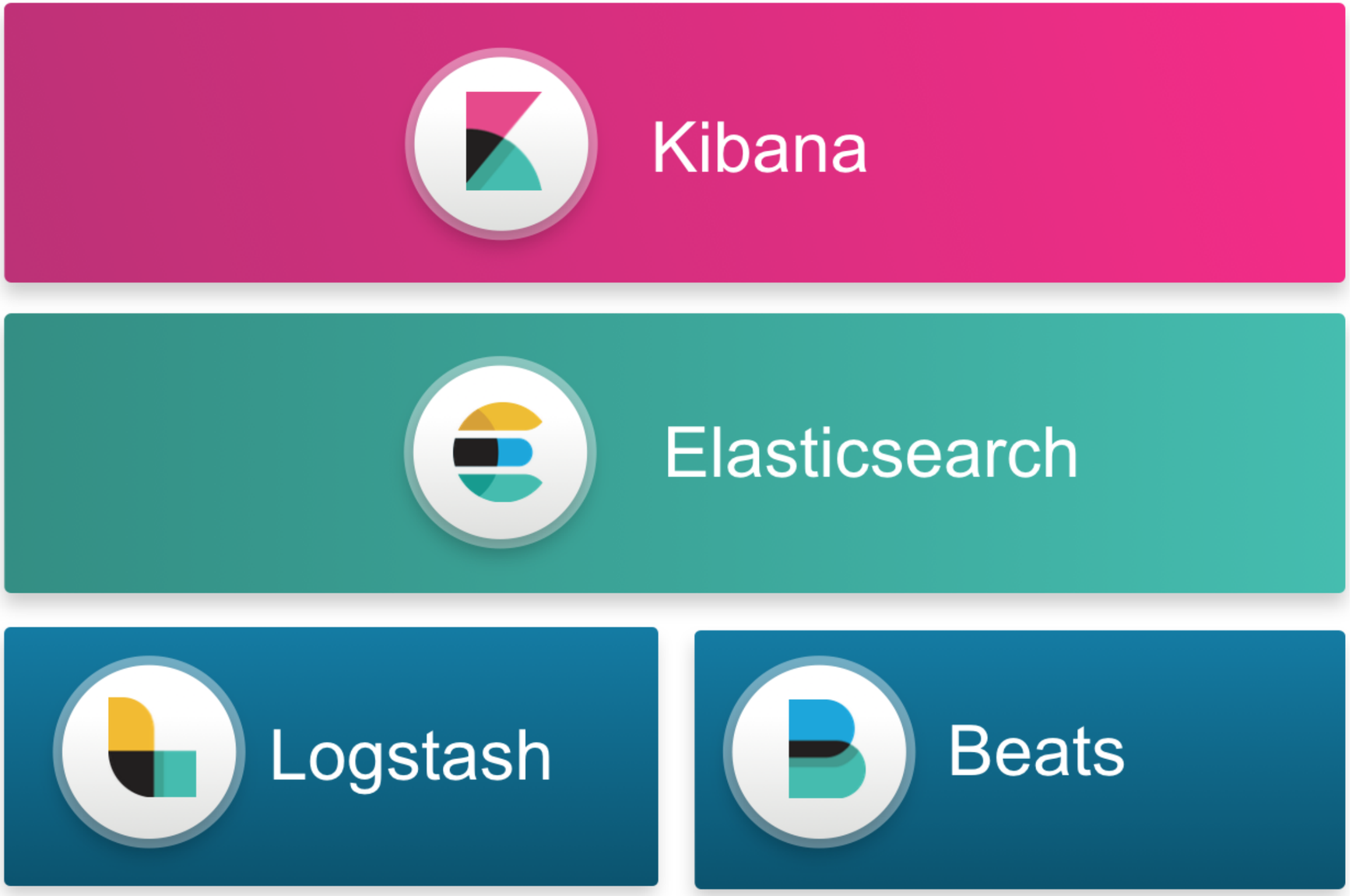


elastic

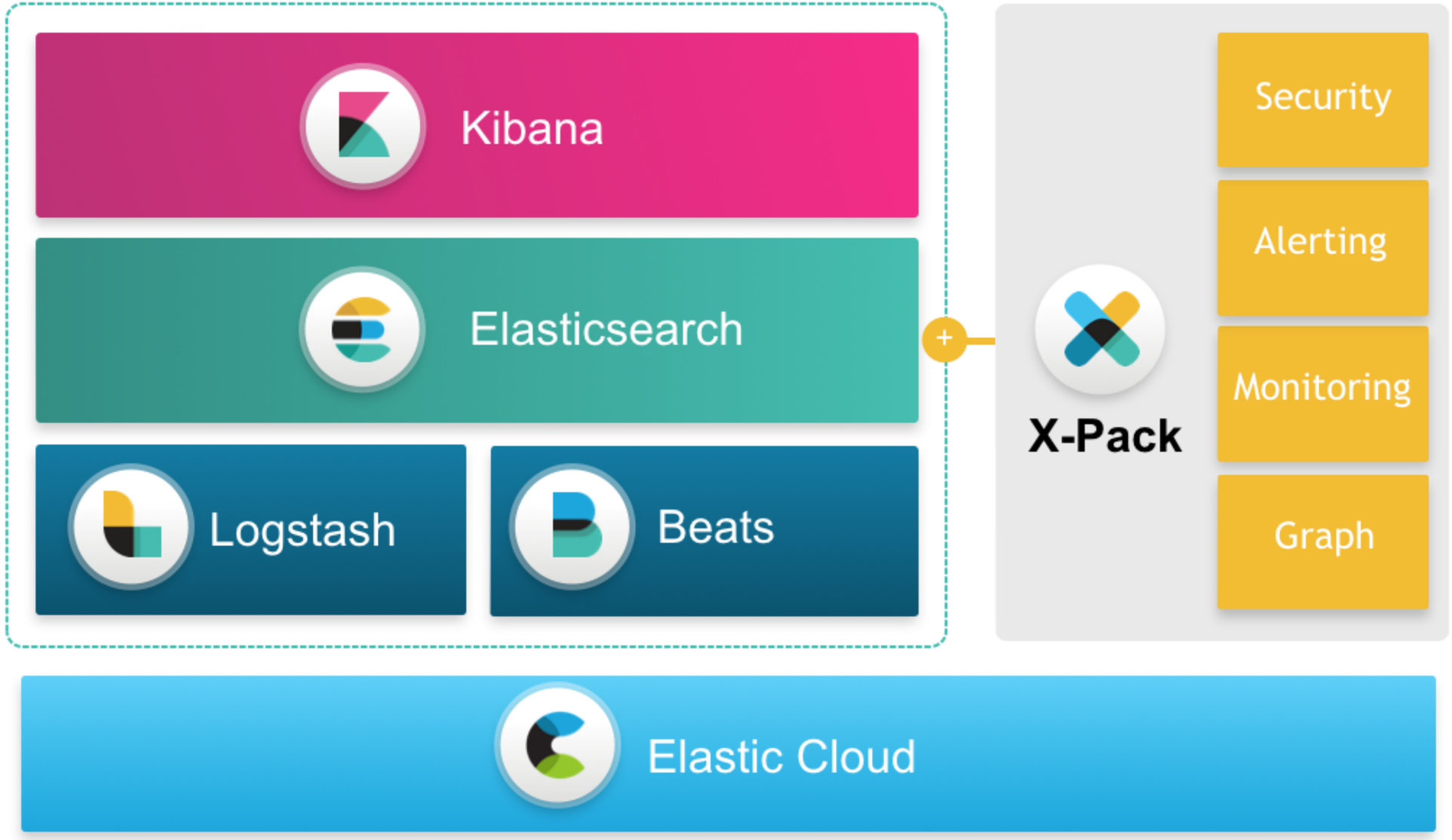
Present

Elastic Stack

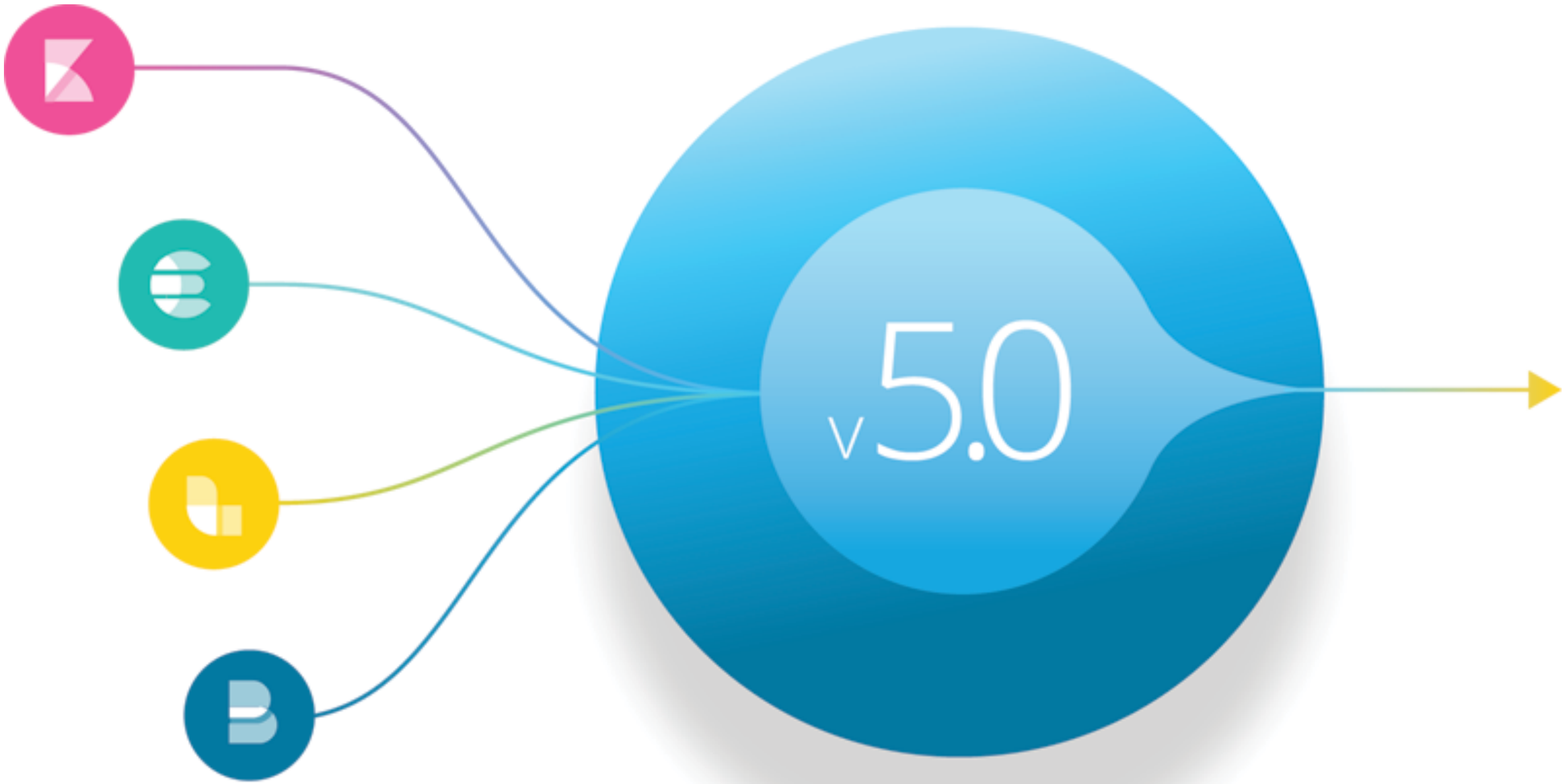
elastic


Kibana
Elasticsearch
Logstash
Beats

Business model


Kibana
Elasticsearch
Logstash
Beats
+
X-Pack
Security
Alerting
Monitoring
Graph
Elastic Cloud

Future


v5.0

Ingest Node

cloud

Enterprise

elastic

Hands-on

elastic

Starting point

https://github.com/xeraa/vagrant-elastic-stack/tree/v5

USB stick

VirtualBox
Box

VirtualBox

Windows, Mac: USB stick
Linux: https://www.virtualbox.org/wiki/Linux_Downloads

Box

Vagrant Ansible Provisioner
o_install.yml

Credentials

`vagrant`

`vagrant`

SSH

```
$ ssh vagrant@127.0.0.1 -p 2222 -o
PreferredAuthentications=password
```

Windows: http://www.putty.org

Elasticsearch

```
$ ansible-playbook 1_configure-elasticsearch.yml
```

Kibana

```
$ ansible-playbook 2_configure-kibana.yml
```

Console

Formerly Sense

Overview

```
GET /
```

```
GET /_cat?v
```

```
GET /_cat/shards?v
```

Insert data

```
PUT /movies

PUT /movies/movie/1
{
  "title": "The Godfather",
  "director": "Francis Ford Coppola",
  "year": 1972
}

GET /movies/movie/1

GET /movies/_mapping
```

Replace data

```
PUT /movies/movie/1
{
  "title": "The Godfather",
  "director": "Francis Ford Coppola",
  "year": 1972,
  "genres": ["Crime", "Drama"]
}

GET /movies/movie/1
```

More data

```
PUT /movies/movie/2
{
  "title": "Lawrence of Arabia",
  "director": "David Lean",
  "year": 1962,
  "genres": ["Adventure", "Biography", "Drama"]
}

PUT /movies/movie/3
{
  "title": "Apocalypse Now",
  "director": "Francis Ford Coppola",
  "year": 1979,
  "genres": ["Drama", "War"]
}
```

Query endpoints

```
/_search
/movies/_search
/movies/movie/_search
```

Queries

```
GET /movies/_search?q=ford
```

```
GET /movies/_search?q=franc*
```

```
GET /movies/_search?q=copola~
```

Queries

```
POST /movies/_search
{
    "query": {
        "query_string": {
            "query": "ford"
        }
    }
}
```

Document score

```
score(q,d) =
            queryNorm(q)
          * coord(q, d)
          * SUM (
                tf(t in d),
                idf(t)²,
                t.getBoost(),
                norm(t, d)
            ) (t in q)
```

https://www.elastic.co/guide/en/elasticsearch/guide/current/scoring-theory.html

Filter

```
POST /movies/_search
{
  "query": {
    "bool": {
      "filter": {
        "term": {
          "year": 1972
        }
      }
    }
  }
}
```

Cleanup

```
DELETE /movies
```

```
GET /movies/movie/1
```

Insert test data

```
$ java -jar /elastic-stack/injector-5.0.jar 100000 1000
```

Overview

```
GET /_cat/shards?v
```

```
GET /_cat/indices/person?v
```

Background

Cluster, node, index, shard, replica

Search

```
GET /person/person/_search
{
  "query": {
    "match": {
      "address.country": "Germany"
    }
  }
}
```

More complex search

```
GET /person/person/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "address.country": "Germany"
          }
        },
        {
          "range": {
            "dateOfBirth": {
              "from": "1970",
              "to": "1971"
            }
          }
        }
      ]
    }
  }
}
```

Aggregation

```
GET /person/person/_search
{
  "size": 0,
  "aggs": {
    "by_country": {
      "terms": {
        "field": "address.country"
      }
    }
  }
}
```

Configure the index pattern

Index name `person`
Time-field name `dateOfBirth`

Kibana Discover

Kibana Visualize

Vertical bar chart with a date histogram
Save

elastic

Kibana Visualize

Pie chart split on the gender
Save

elastic

Kibana Visualize

Pie chart split on the country and then city
Save

elastic

Kibana Visualize

Tile map
Save

elastic

Kibana Dashboard

Combine all the saved visualizations

Logstash

```
$ ansible-playbook 3_configure-logstash.yml
```

Beats

```
$ ansible-playbook 4_configure-filebeat.yml
$ ansible-playbook 4_configure-metricbeat.yml
$ ansible-playbook 4_configure-packetbeat.yml
```

Add nginx logs

/var/log/nginx/access.log

Filebeat configuration

$ sudo vi /etc/filebeat/filebeat.yml

```
filebeat:
  prospectors:
    - input_type: log
      paths:
        - /var/log/*.log
        - /var/log/syslog

    - input_type: log
      paths:
        - /var/log/nginx/access.log
      document_type: nginx-access
```

Filebeat restart

```
$ sudo service filebeat restart
```

Logstash pattern

```
$ sudo mkdir -p /opt/logstash/patterns
$ sudo tee -a /opt/logstash/patterns/nginx >/dev/null <<'EOF'
NGUSERNAME [a-zA-Z\.\@\-\+_%]+
NGUSER %{NGUSERNAME}
NGINXACCESS %{IPORHOST:clientip} %{NGUSER:ident} %{NGUSER:auth}
            \[%{HTTPDATE:timestamp}\] "%{WORD:verb}
            %{URIPATHPARAM:request} HTTP/%{NUMBER:httpversion}"
            %{NUMBER:response} (?:%{NUMBER:bytes}|-)
            (?:"(?:%{URI:referrer}|-)"|%{QS:referrer}) %{QS:agent}
EOF
$ sudo chown -R logstash:logstash /opt/logstash/patterns/
```

Without the linebreaks in `NGINXACCESS`

Logstash filter

```
$ sudo tee -a /etc/logstash/conf.d/11-nginx-filter.conf >/dev/null <<'EOF'
filter {
  if [type] == "nginx-access" {
    grok {
      match => { "message" => "%{NGINXACCESS}" }
    }
  }
}
EOF
$ sudo service logstash restart
```

Debug Logstash

```
$ tail /var/log/logstash/logstash.err
$ tail /var/log/logstash/logstash.log
```

Kibana Discover

`_type: nginx-access`

Kibana Visualize

Line chart: `filebeat-*`
X-Axis: Date Histogram
Y-Axis: Count

Kibana Visualize

Add sub-buckets
Split lines and move up
Aggregation: Terms
Field: type

Beats dashboards

```
$ ansible-playbook 5_configure-dashboards.yml
```

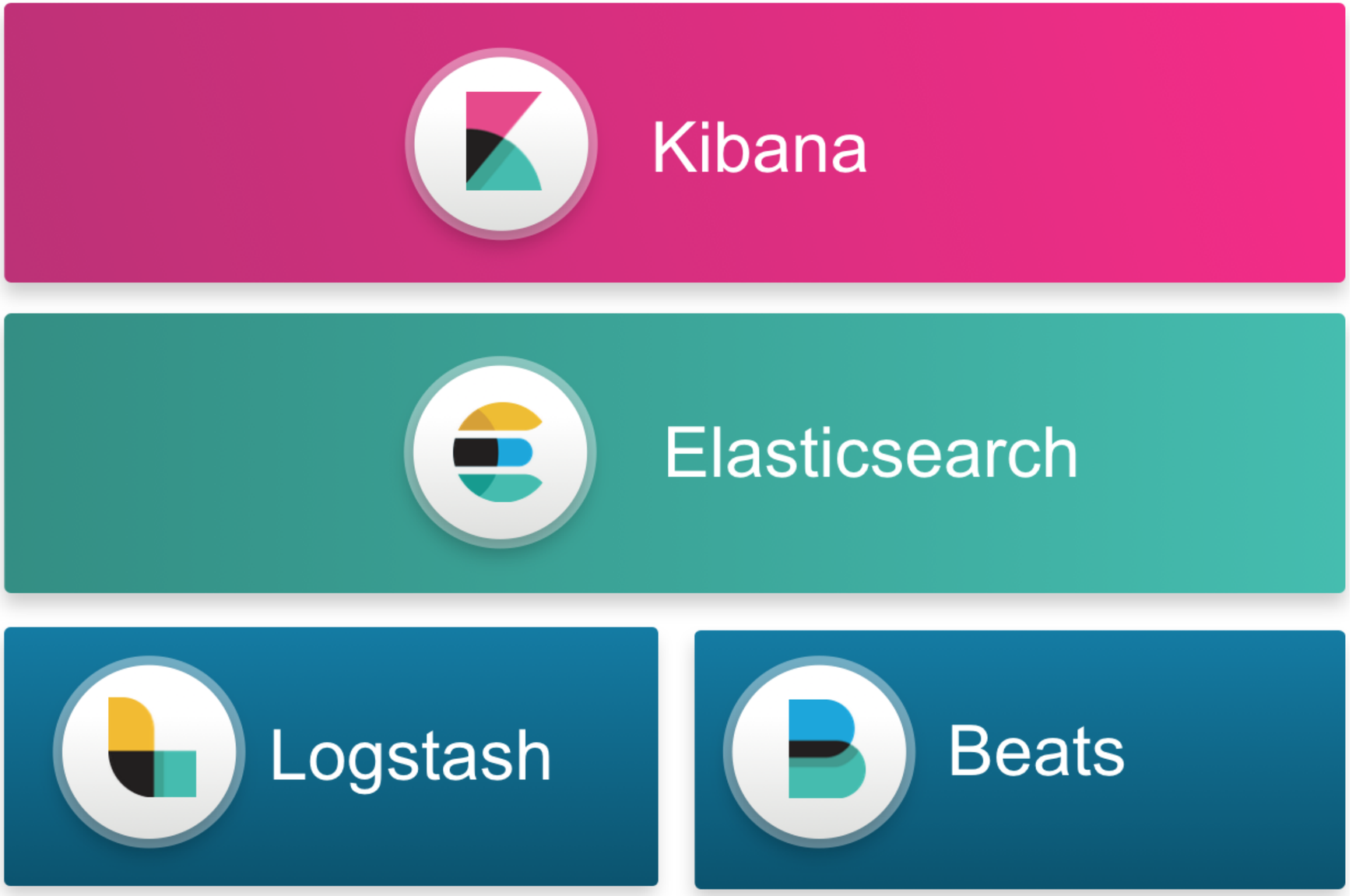
Plugins

```
$ ansible-playbook 6_add-plugins.yml
```

Timelion

```
.es(*)

.es(index=filebeat*)

.es(index=filebeat*), .es(index=metricbeat*), .es(index=packetbeat*)

.es(index=filebeat*).label("file"), .es(index=metricbeat*).label("metric"),
.es(index=packetbeat*).label("packet")

(.es(index=filebeat*).label("file"), .es(index=metricbeat*).label("metric"),
 .es(index=packetbeat*).label("packet")).cusum()

.wbi(FR), .wbi(DE)

(.wbi(FR), .wbi(DE)).derivative()
```

Conclusion


Kibana
Elasticsearch
Logstash
Beats

Want more?

https://www.elastic.co/training

Questions

Thanks!

Philipp Krenn @xeraa

PS: Stickers