TITANIUM SPONSORS

epiq

PAIGE TECHNOLOGIES
INTELLIGENT PAIRING. PERPETUAL SUCCESS.

ADAPTIVE™
SOLUTIONS GROUP

VALOREM
CONSULTING

Platinum Sponsors

VinSolutions
Make every connection count.

jack henry & ASSOCIATES INC.®

KU
EDWARDS CAMPUS
The University of Kansas

NAIC
National Association of Insurance Commissioners

NIPR
NATIONAL INSURANCE PRODUCER REGISTRY

JET BRAINS

Stormpath

TEKsystems
Our people make IT possible

stackify

Octopus Deploy

Gold Sponsors

OAKWOOD

KEYHOLE SOFTWARE

Cerner

PocketCake

Bradford & Galt

GARMIN.

dsi

PLURALSIGHT

Commerce Bank

OBJECT PARTNERS

DST SYSTEMS

Advantage Tech

AUREUS GROUP
Finance | Systems | Executive

Balance Innovations

LRS Consulting Services

SPARKPOST

twilio

UNITEDLEX

StrongLoop
An IBM Company

Useagility

TRIPLE-I

PeopleAdmin

smg service management group®

NERDERY

wirestream™
enabling ideas

RAYGUN

VML

Children's Mercy KANSAS CITY

CENTRIQ TRAINING

mylo™
POWERED BY LOCKTON

SAIC
Redefining Ingenuity™

H&R BLOCK

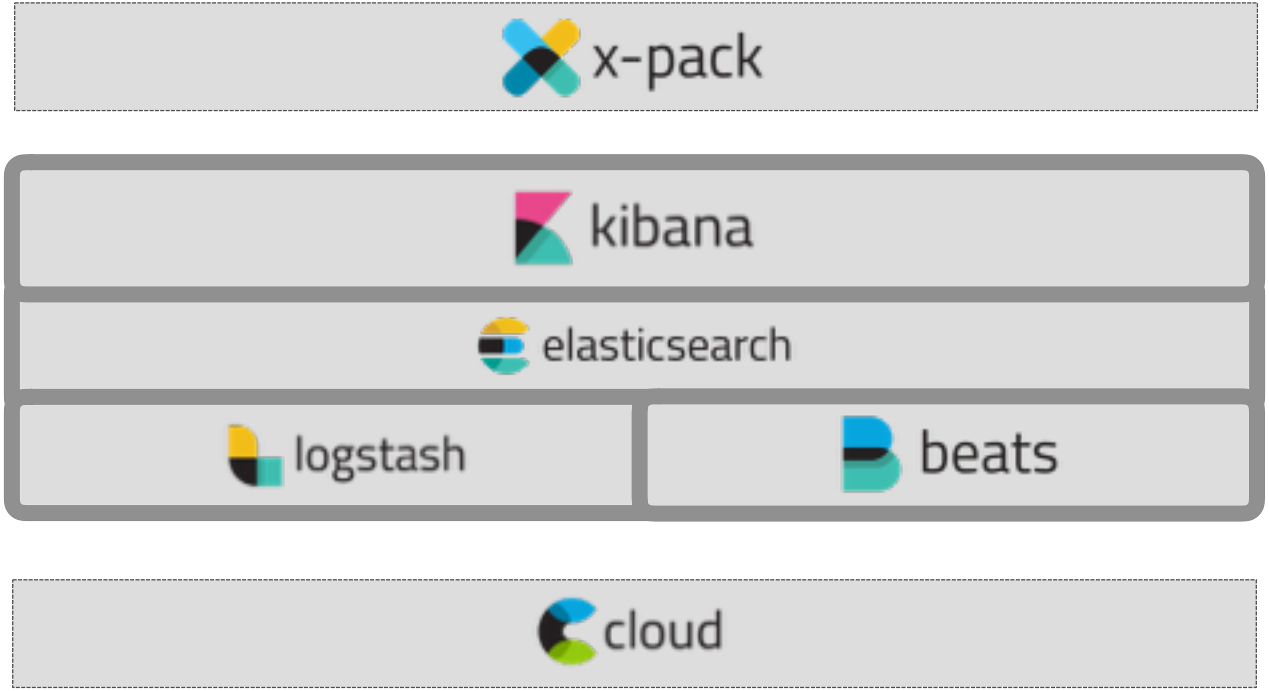
ca technologies

elastic

Elasticsearch for SQL users

Shaunak Kashyap
Developer at Elastic
@shaunak

The Elastic Stack



Plugins

User Interface

Store, Index & Analyze

Ingest

Hosted Service

Agenda

1 Search queries

2 Data modeling

3 Architecture

Agenda

1 Search queries

2 Data modeling

3 Architecture

Agenda

1 Search queries

2 Data modeling

3 Architecture

https://www.flickr.com/photos/samhames/4422128094

Search Queries

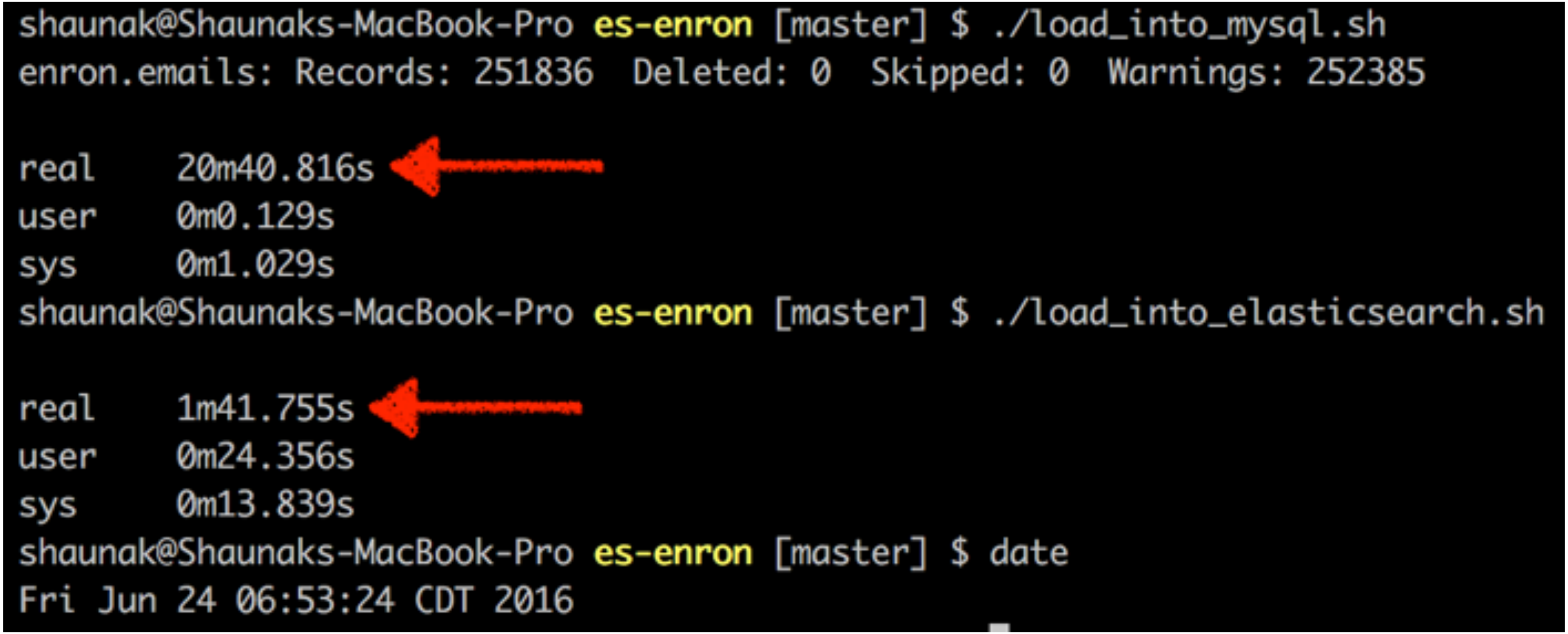
Schemas

```
CREATE TABLE IF NOT EXISTS emails (
  sender        VARCHAR(255) NOT NULL,
  recipients    TEXT,
  cc            TEXT,
  bcc           TEXT,
  subject       VARCHAR(1024),
  body          MEDIUMTEXT,
  datetime      DATETIME
);

CREATE INDEX emails_sender ON emails(sender);
CREATE FULLTEXT INDEX emails_subject ON emails(subject);
CREATE FULLTEXT INDEX emails_body ON emails(body);
```

```
curl -XPUT 'http://localhost:9200/enron' -d'
{
  "mappings": {
    "email": {
      "properties": {
        "sender": { "type": "keyword" },
        "recipients": { "type": "keyword" },
        "cc": { "type": "keyword" },
        "bcc": { "type": "keyword" },
        "subject": { "type": "text", "analyzer": "english" },
        "body": { "type": "text", "analyzer": "english" }
      }
    }
  }
```

Loading the data



```
shaunak@Shaunaks-MacBook-Pro es-enron [master] $ ./load_into_mysql.sh
enron.emails: Records: 251836  Deleted: 0  Skipped: 0  Warnings: 252385

real	20m40.816s
user	0m0.129s
sys	0m1.029s
shaunak@Shaunaks-MacBook-Pro es-enron [master] $ ./load_into_elasticsearch.sh

real	1m41.755s
user	0m24.356s
sys	0m13.839s
shaunak@Shaunaks-MacBook-Pro es-enron [master] $ date
Fri Jun 24 06:53:24 CDT 2016
```

[LIVE DEMO]

- Search for text in a single field
- Search for text in multiple fields
- Search for a phrase

https://github.com/ycombinator/es-enron

Other Search Features

Stemming

- Jump, jumped, jumping

Synonyms

- Queen, monarch

Did you mean?

- Monetery => Monetary

https://www.flickr.com/photos/ericparker/7854157310

Data Modeling

To analyze (`text`) or not to analyze (`keyword`)?

```
PUT cities/city/1
{
  "city": "Omaha",
  "population": 434353
}
```

```
PUT cities/city/2
{
  "city": "New Albany",
  "population": 8829
}
```

```
PUT cities/city/3
{
  "city": "New York",
  "population": 8406000
}
```

+

QUERY

```
POST cities/_search
{
  "query": {
    "match": {
      "city": "New Albany"
    }
  }
}
```

= ?

To analyze (`text`) or not to analyze (`keyword`)?

```
PUT cities/city/1
{
  "city": "Omaha",
  "population": 434353
}
```

```
PUT cities/city/2
{
  "city": "New Albany",
  "population": 8829
}
```

```
PUT cities/city/3
{
  "city": "New York",
  "population": 8406000
}
```

Term	Document IDs
albany	2
new	2,3
omaha	1
york	3

To analyze (`text`) or not to analyze (`keyword`)?

MAPPING

```
PUT cities
{
  "mappings": {
    "city": {
      "properties": {
        "city": {
          "type": "keyword"
        }
      }
    }
  }
}
```

Term	Document IDs
New Albany	2
New York	3
Omaha	1

Relationships: Application-side joins

```
PUT blog/author/1
{
  "name": "John Doe",
  "bio": "..."
}
```

```
PUT blog/post/1
{
}
```

```
PUT blog/post/2
{
}
```

```
PUT blog/post/3
{
  "author_id": 1,
  "title": "...",
  "body": "..."
}
```

QUERY 1

```
POST blog/author/_search
{
  "query": {
    "match": {
      "name": "John"
    }
  }
}
```

QUERY 2

```
POST blog/post/_search
{
  "query": {
    "match": {
      "author_id": <each id from query 1 result>
    }
  }
}
```

Relationships: Data denormalization

```
PUT blog/post/1
{
}
```

```
PUT blog/post/2
{
}
```

```
PUT blog/post/3
{
  "author_name": "John Doe",
  "title": "...",
  "body": "..."
}
```

QUERY

```
POST blog/post/_search
{
  "query": {
    "match": {
      "author_name": "John"
    }
  }
}
```

Relationships: Nested objects

```
PUT blog/author/1
{
  "name": "John Doe",
  "bio": "...",
  "blog_posts": [
    {
      "title": "...",
      "body": "..."
    },
    {
      "title": "...",
      "body": "..."
    },
    {
      "title": "...",
      "body": "..."
    }
  ]
}
```

QUERY

```
POST blog/author/_search
{
  "query": {
    "match": {
      "name": "John"
    }
  }
}
```

Relationships: Parent-child documents

```
PUT blog
{
  "mappings": {
    "author": {},
    "post": {
      "_parent": {
        "type": "author"
      }
    }
  }
}
```

```
PUT blog/author/1
{
  "name": "John Doe",
  "bio": "..."
}
```

```
PUT blog/post/1?parent=1
{
}
```

```
PUT blog/post/2?parent=1
{
}
```

```
PUT blog/post/3?parent=1
{
  "title": "...",
  "body": "..."
}
```

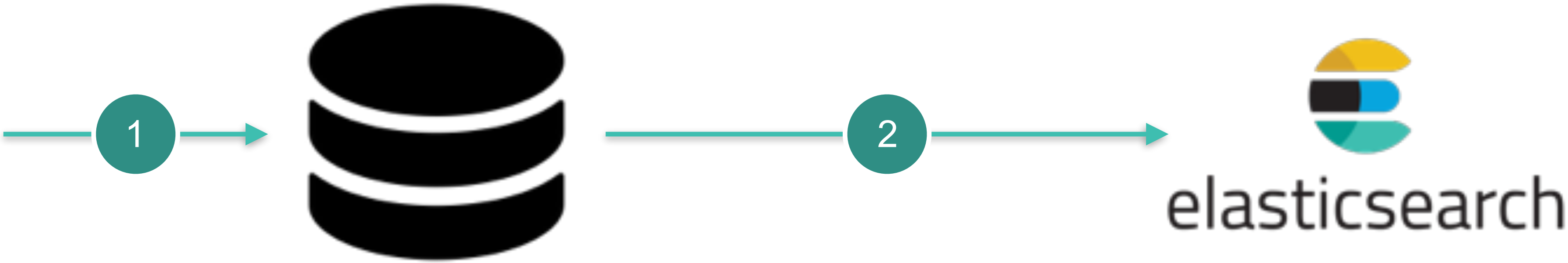
QUERY

```
POST blog/post/_search
{
  "query": {
    "has_parent": {
      "type": "author",
      "query": {
        "match": {
          "name": "John"
      }
    }
  }
}
```

https://www.flickr.com/photos/haribote/4871284379/

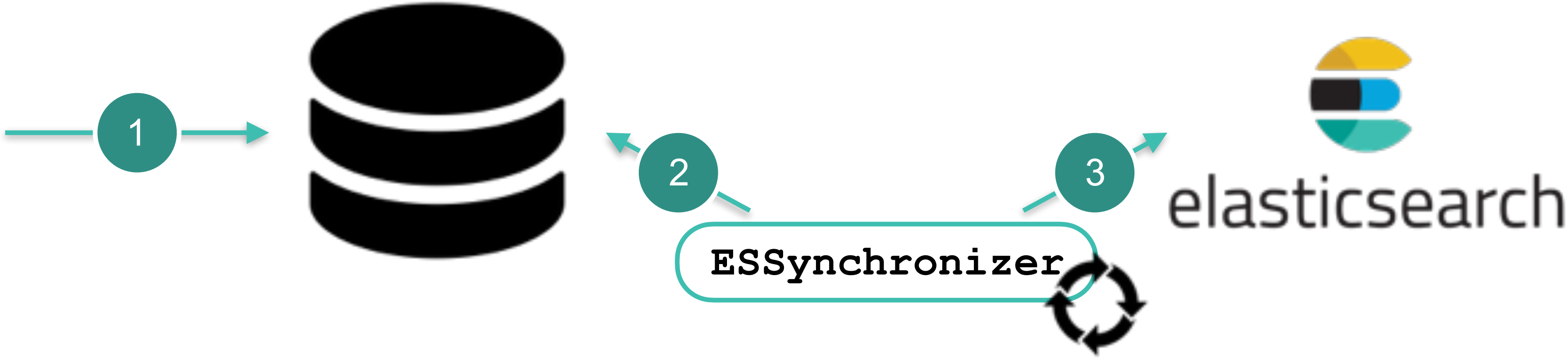
Architecture

RDBMS Triggers



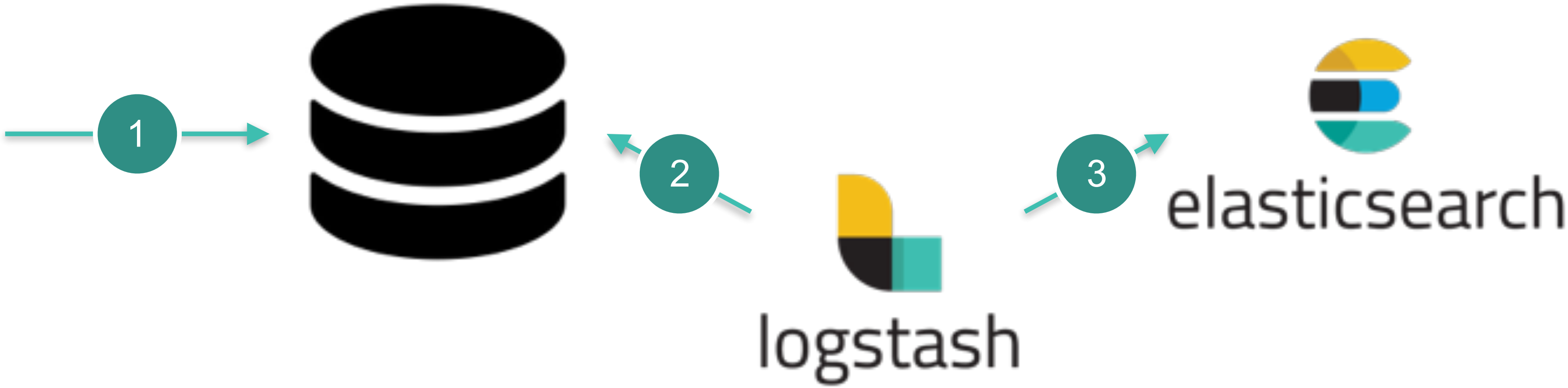
database by Creative Stall from the Noun Project

Async replication to Elasticsearch

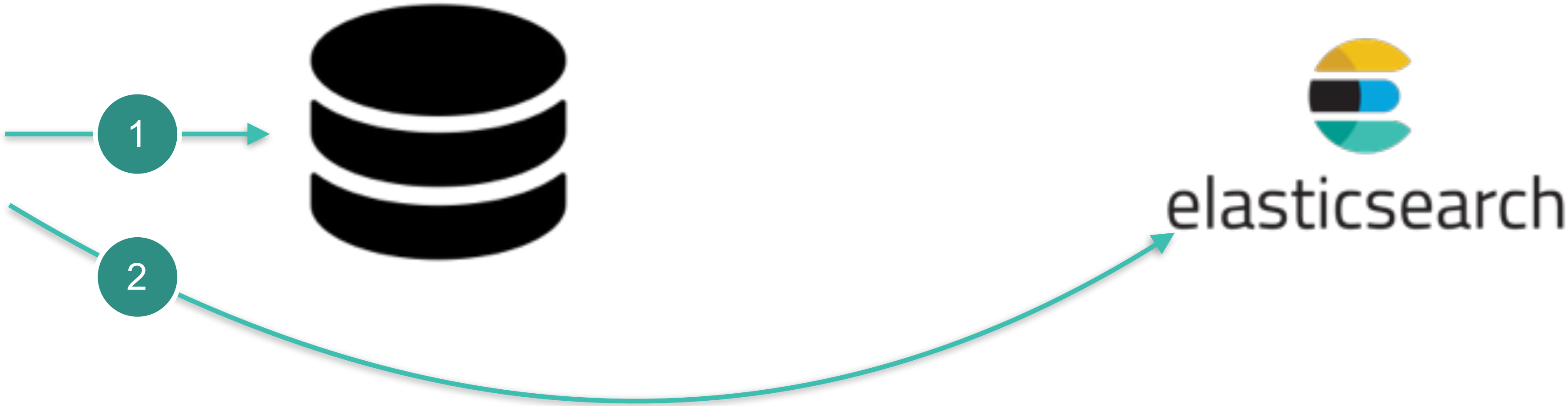


flow by Yamini Ahluwalia from the Noun Project

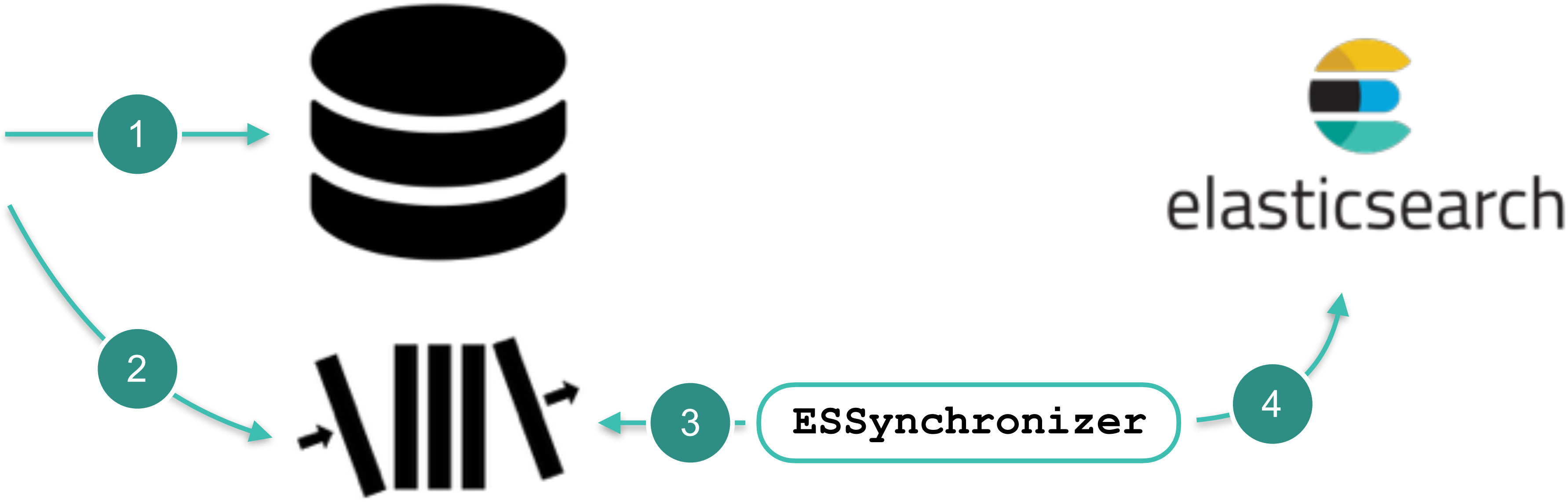
Async replication to Elasticsearch with Logstash



Forked writes from application

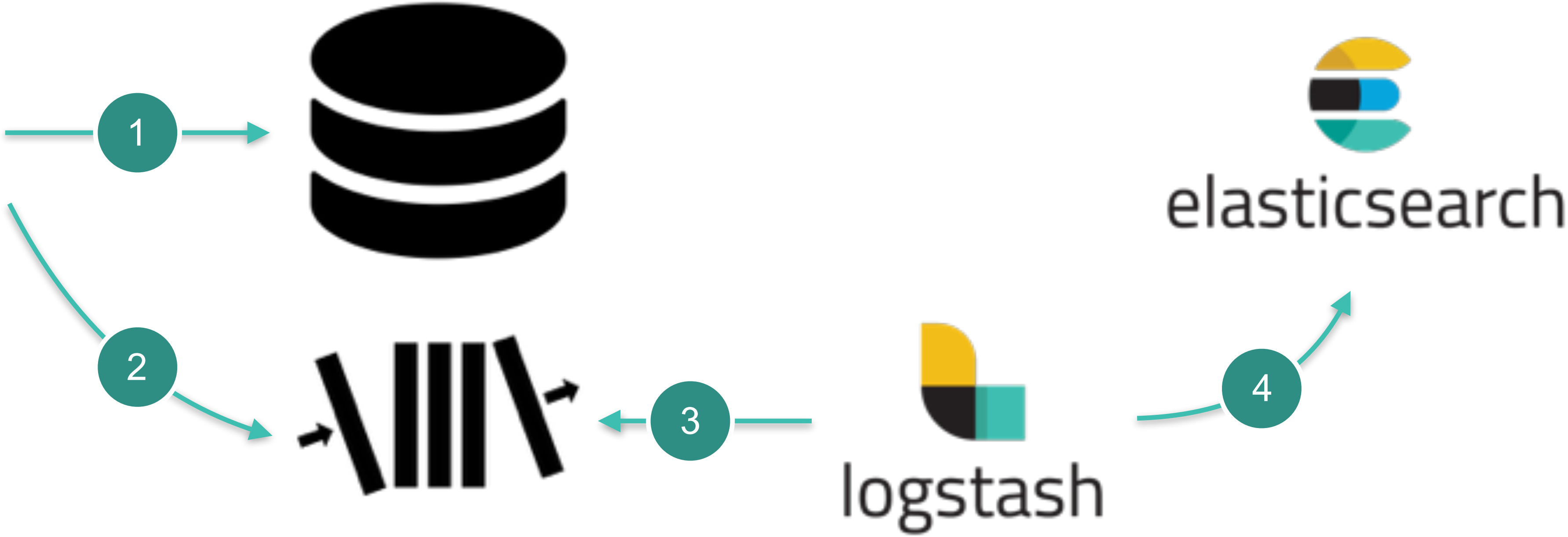


Forked writes from application (more robust)



queue by Huu Nguyen from the Noun Project

Forked writes from application (more robust with Logstash)



https://www.flickr.com/photos/nicknormal/2245559230/

Questions?

@shaunak

elastic