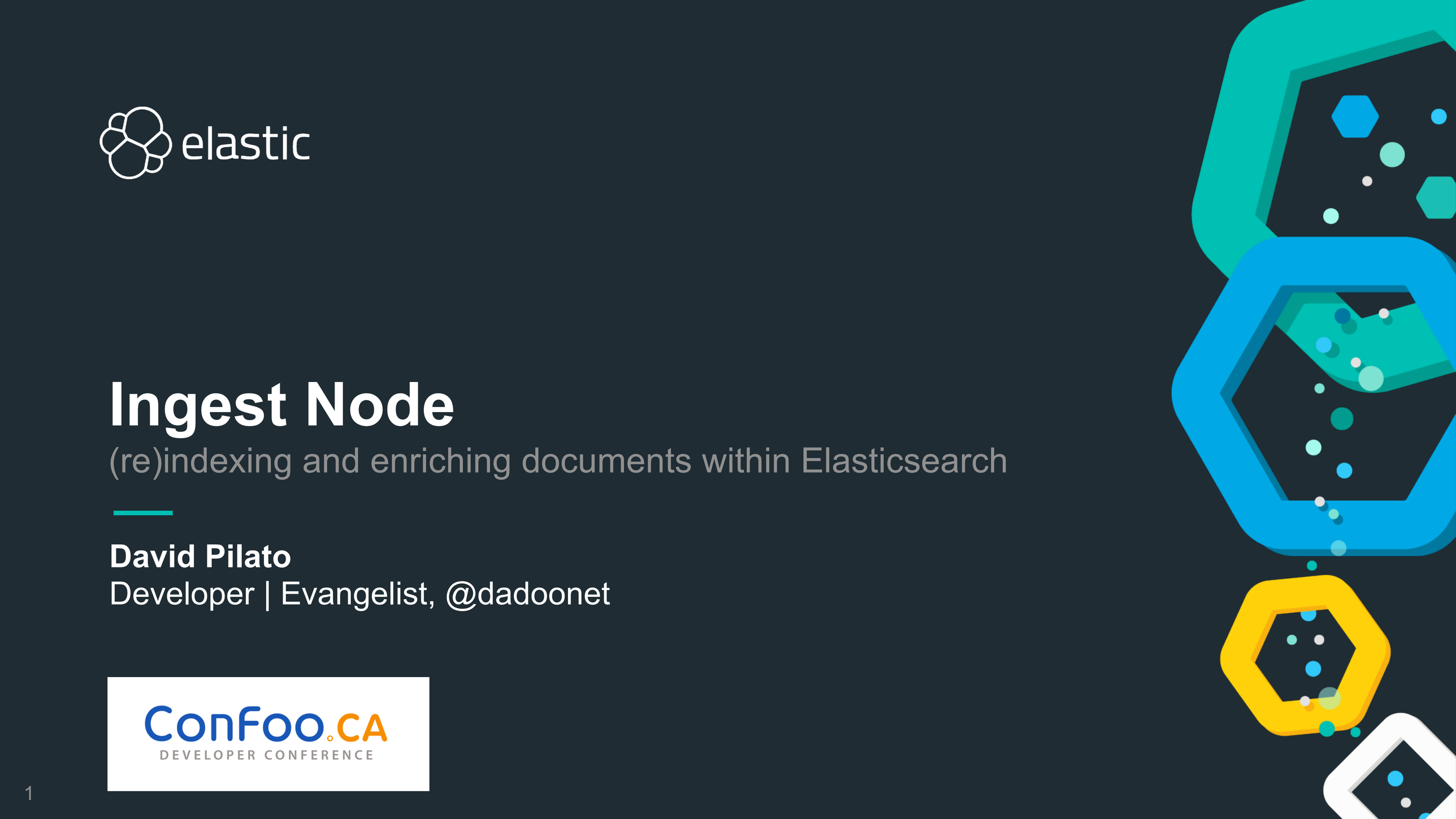


Ingest Node

(re)indexing and enriching documents within Elasticsearch

David Pilato
Developer | Evangelist, @dadoonet

ConFoo.ca
DEVELOPER CONFERENCE

elastic / elasticsearch

Unwatch 2,147 | Unstar 25,529 | Fork 9,006

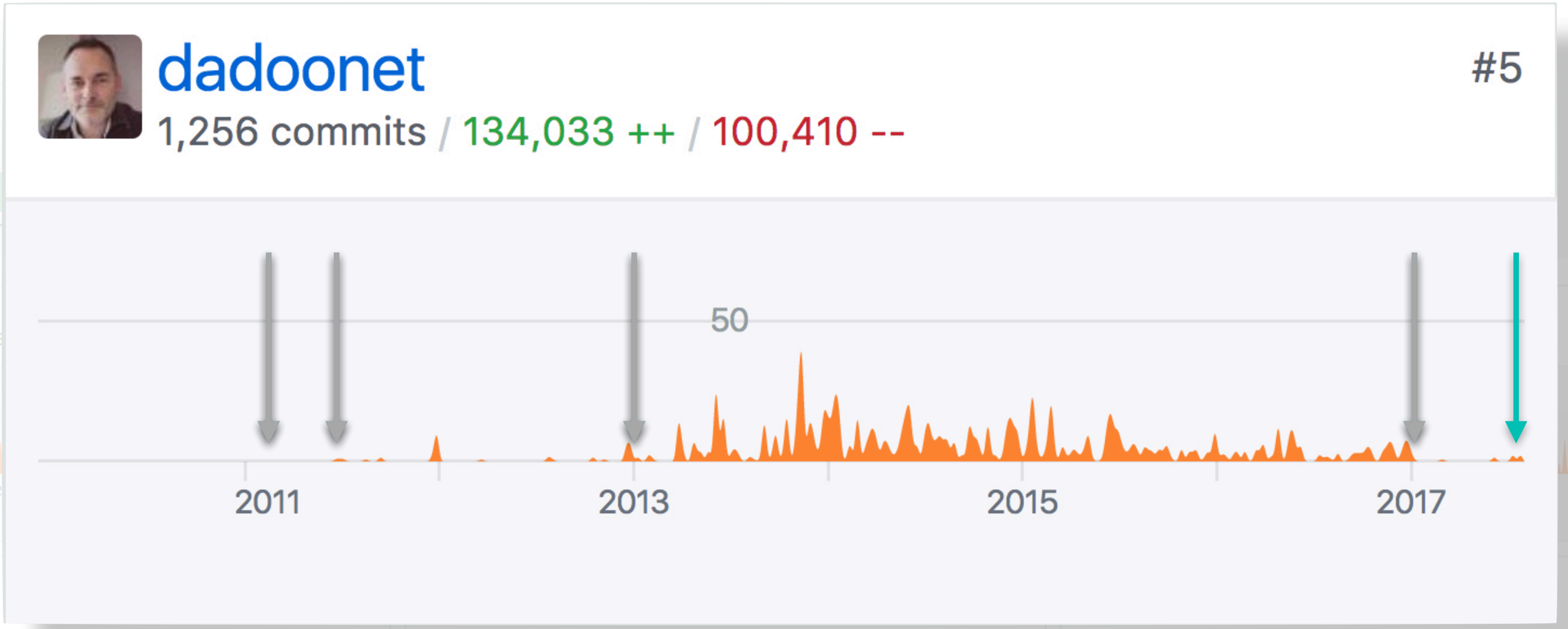
Code | Issues 1,192 | Pull requests 121 | Projects 1 | Settings | Insights

Contributors | Traffic | Commits | Code frequency | Punch card | Network | Members | Dependents

Feb 7, 2010 – Oct 2, 2017

Contributions to master, excluding merge commits

Contributions: Commits



kimchy
4,737 commits / 1,401,53

#3

jasontedor
1,373 commits / 103,792

#6

clintongormley #7

rjernst #8

nik9000 #9

sli.do

Join at

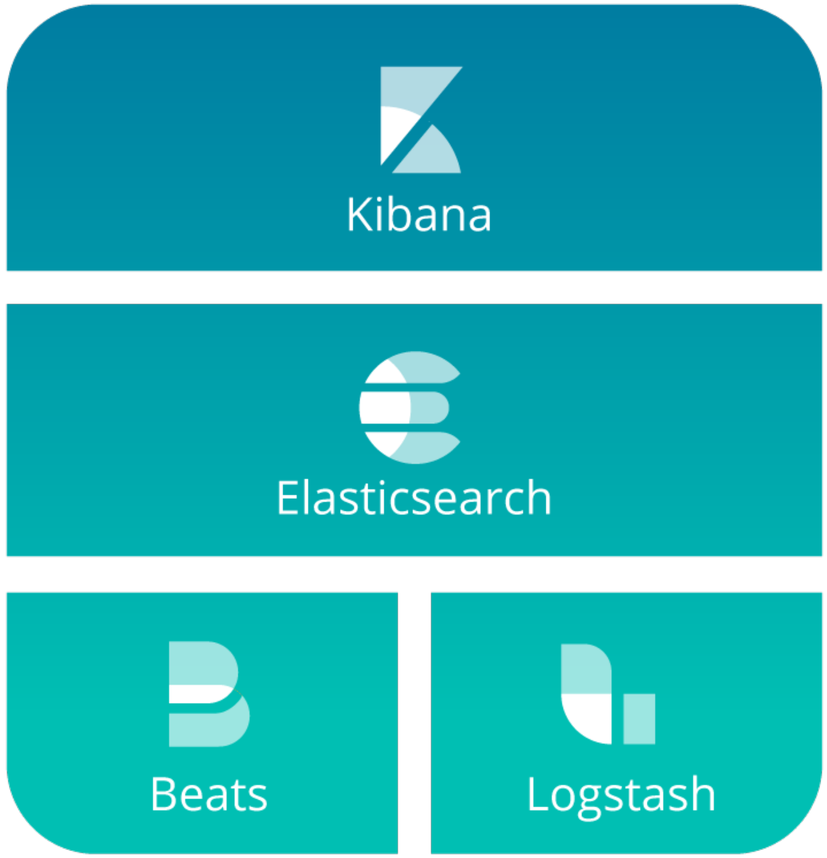
slido.com

#elastic

sli.do/elastic

Elastic Stack

100% open source
No enterprise edition



X-Pack

Single install
Extensions for the Elastic Stack
Subscription pricing

Security

Alerting

Monitoring

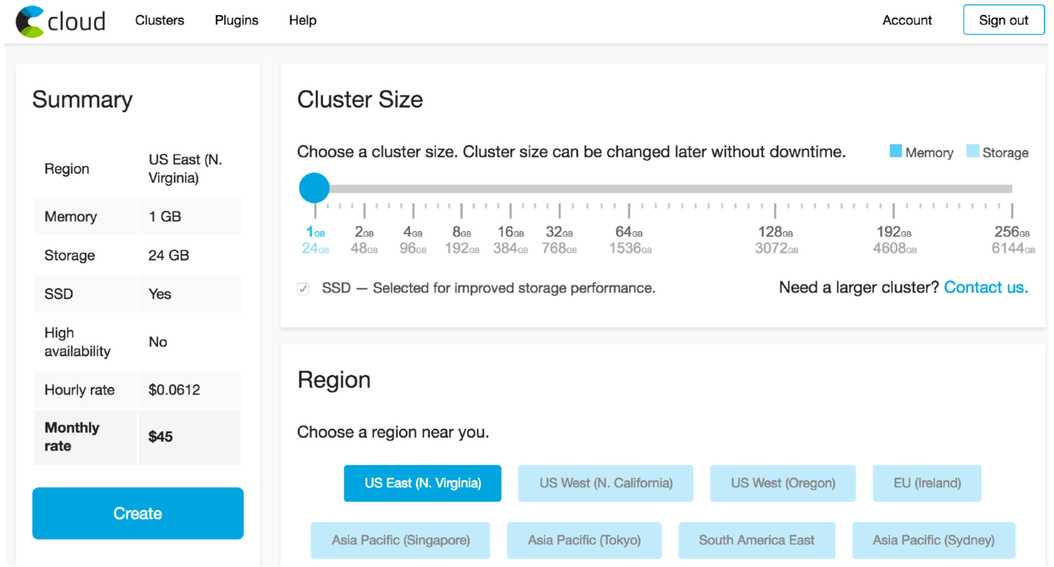
Reporting

Graph

Machine Learning

Elastic Cloud

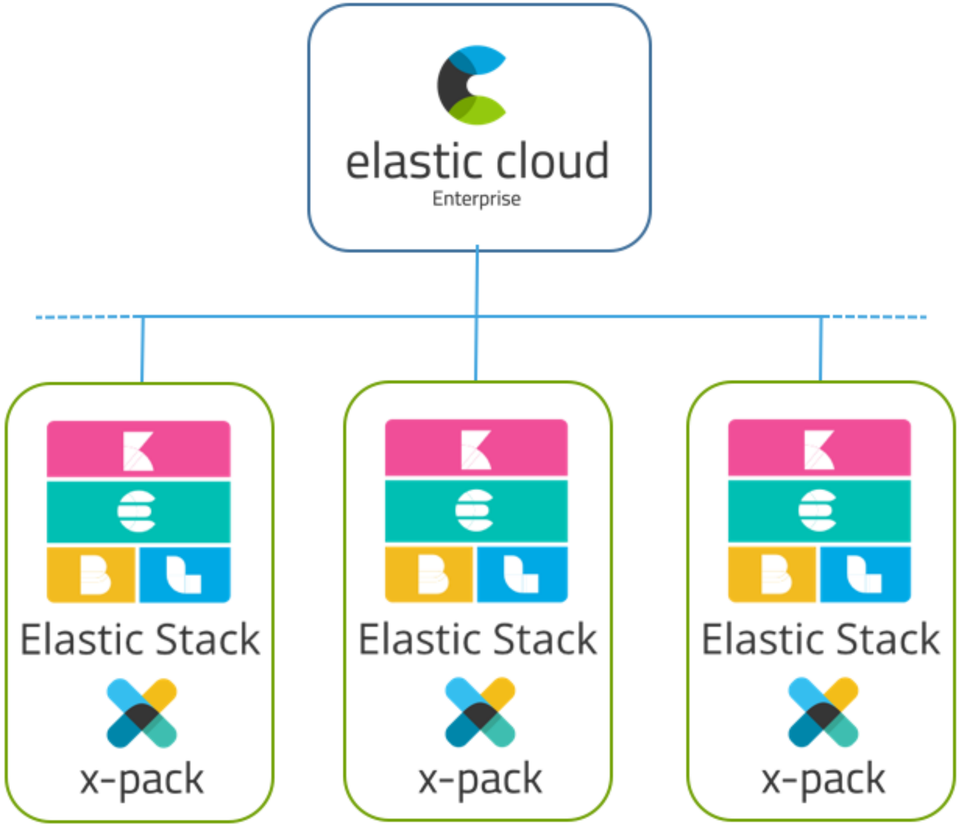
Hosted Elasticsearch & Kibana
Includes X-Pack features
Starts at $45/mo



Available in AWS and Google Cloud Platform

Elastic Cloud Enterprise

Provision and manage multiple Elastic Stack environments; Expose logging as a service to your entire organization



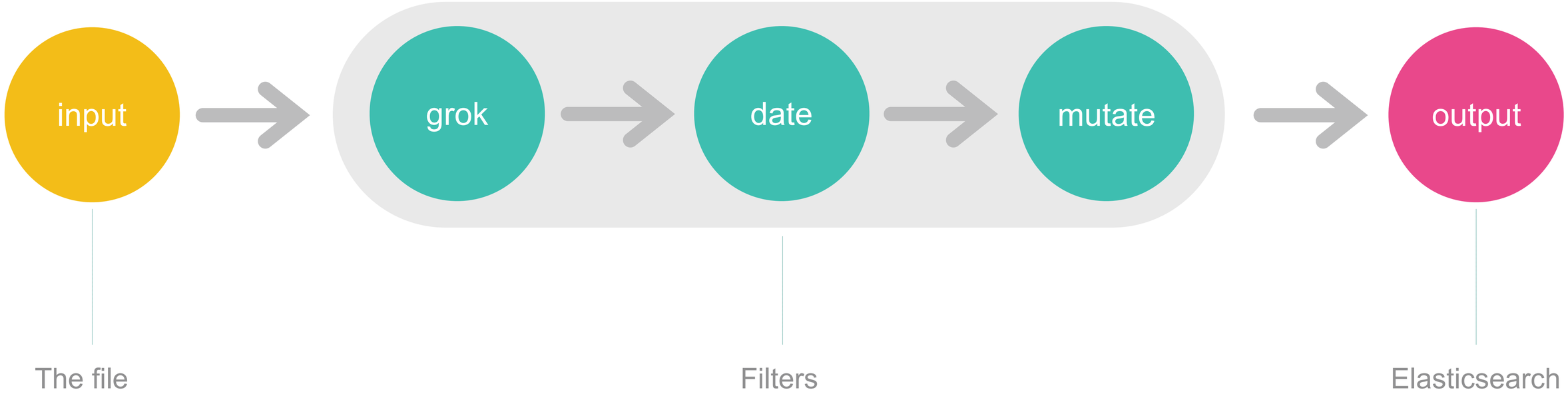
Why ingest node?

elastic

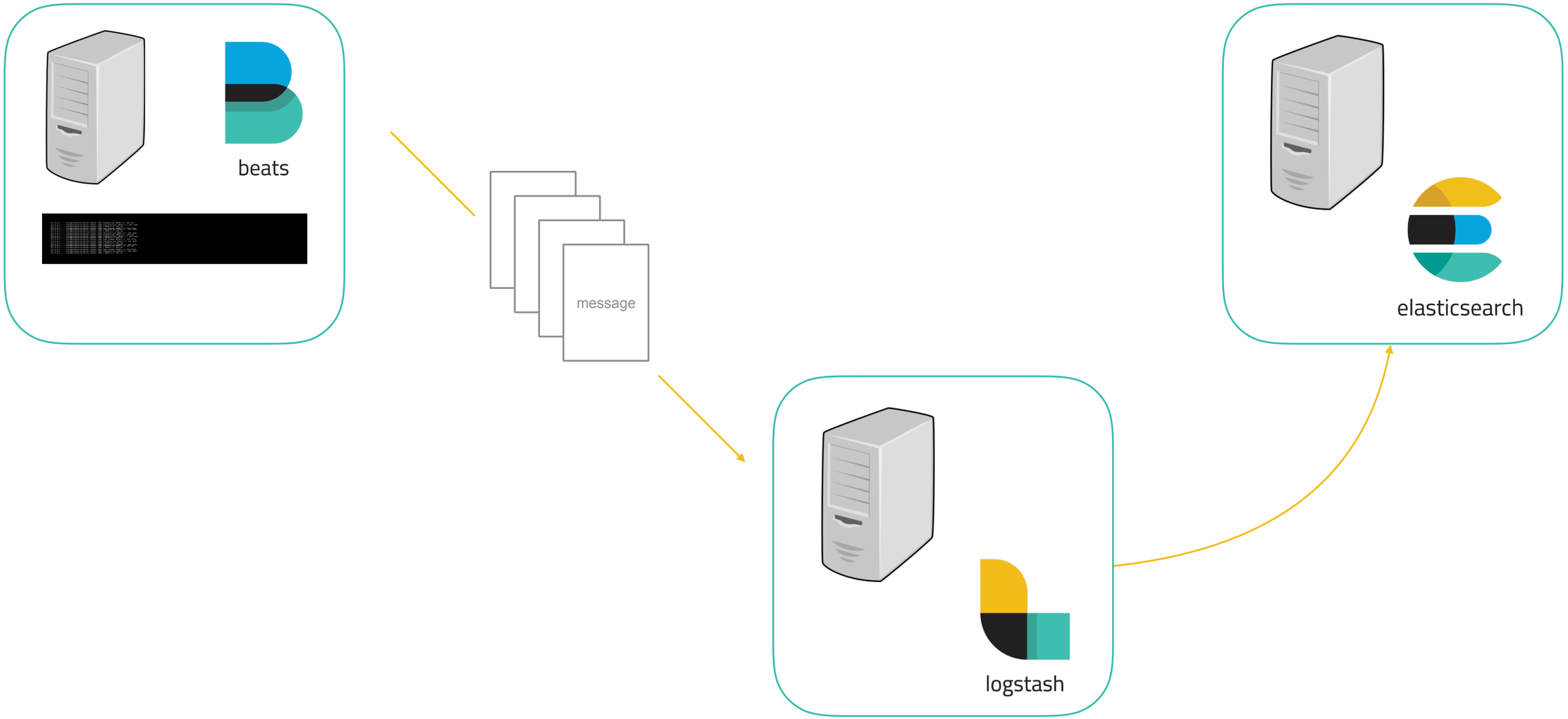
I just want to tail a log file...

Ops Engineer

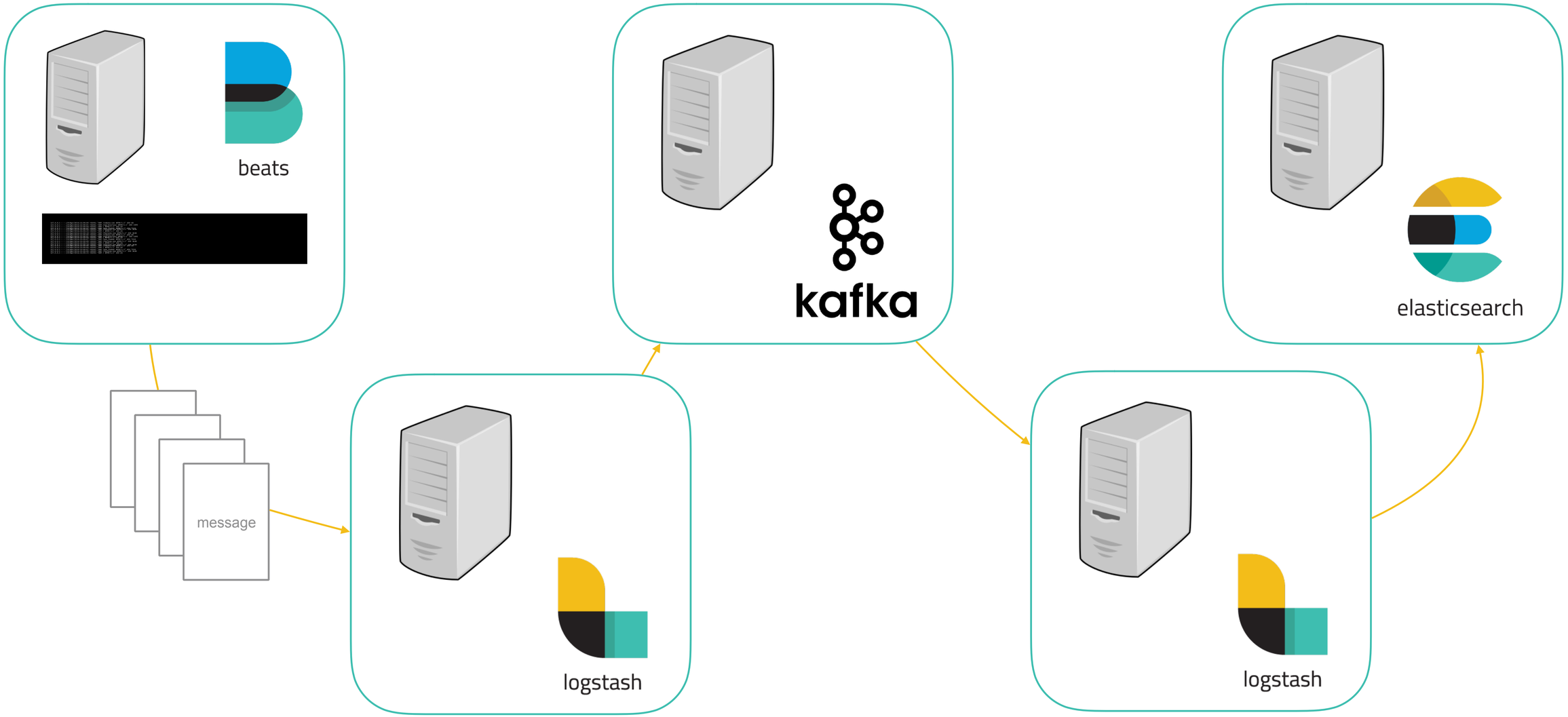
Logstash: collect, enrich & transport



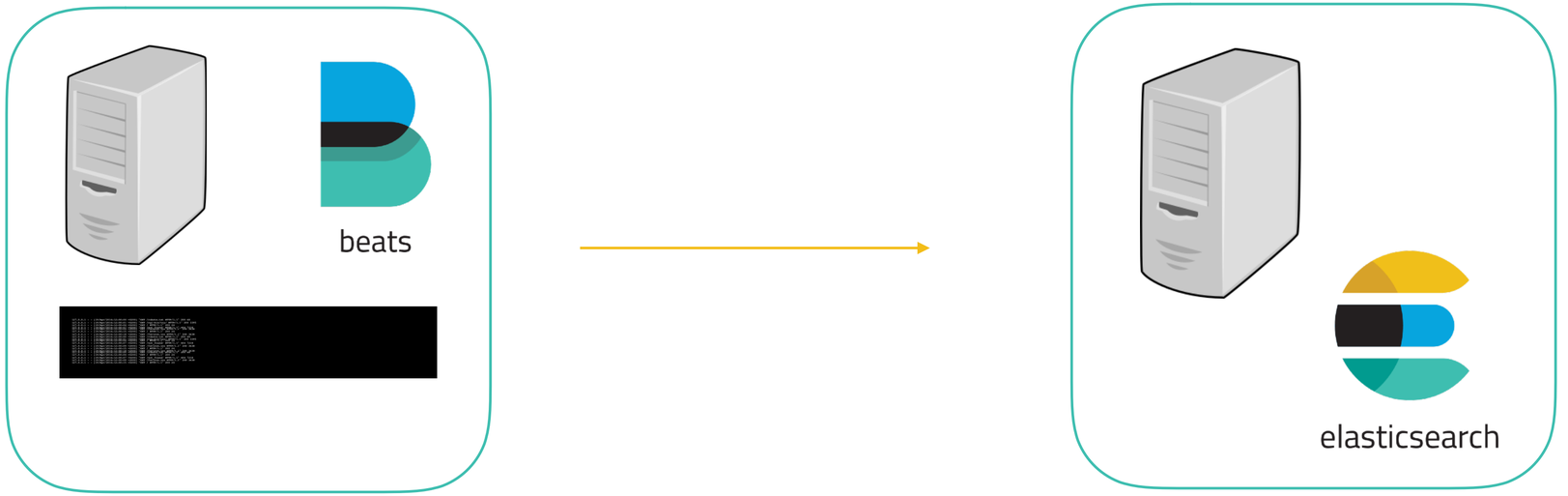
Logstash common setup



Or ...



Ingest node setup



Filebeat: collect and ship

```
127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] "GET / HTTP/1.1" 200 24
127.0.0.1 - - [19/Apr/2016:12:00:07 +0200] "GET /not_found/ HTTP/1.1" 404 7218
127.0.0.1 - - [19/Apr/2016:12:00:09 +2000] "GET /favicon.ico HTTP/1.1" 200 3638
```


```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / HTTP/1.1\" 200 24"
}
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:07 +0200] \"GET /not_found/ HTTP/1.1\" 404 7218"
}
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:09 +2000] \"GET /favicon.ico HTTP/1.1\" 200 3638"
}
```

Elasticsearch: enrich and index

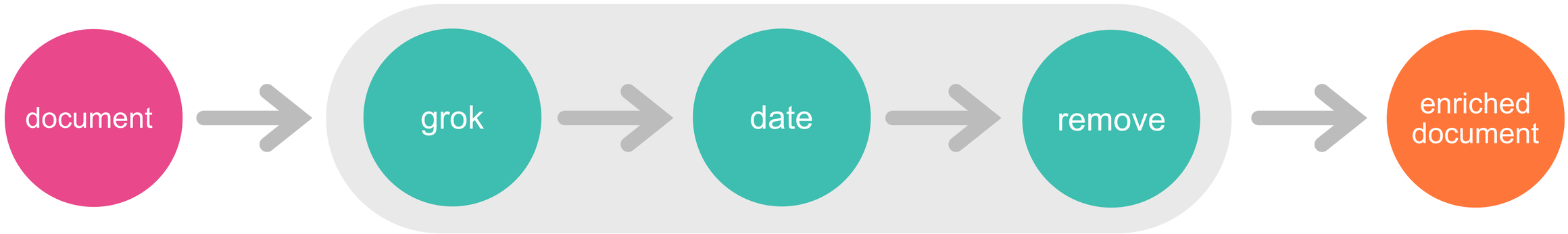
```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / HTTP/1.1\" 200 24"
}
```

```
{
    "request" : "/",
    "auth" : "-",
    "ident" : "-",
    "verb" : "GET",
    "@timestamp" : "2016-04-19T10:00:04.000Z",
    "response" : "200",
    "bytes" : "24",
    "clientip" : "127.0.0.1",
    "httpversion" : "1.1",
    "rawrequest" : null,
    "timestamp" : "19/Apr/2016:12:00:04 +0200"
}
```

How does ingest node work?

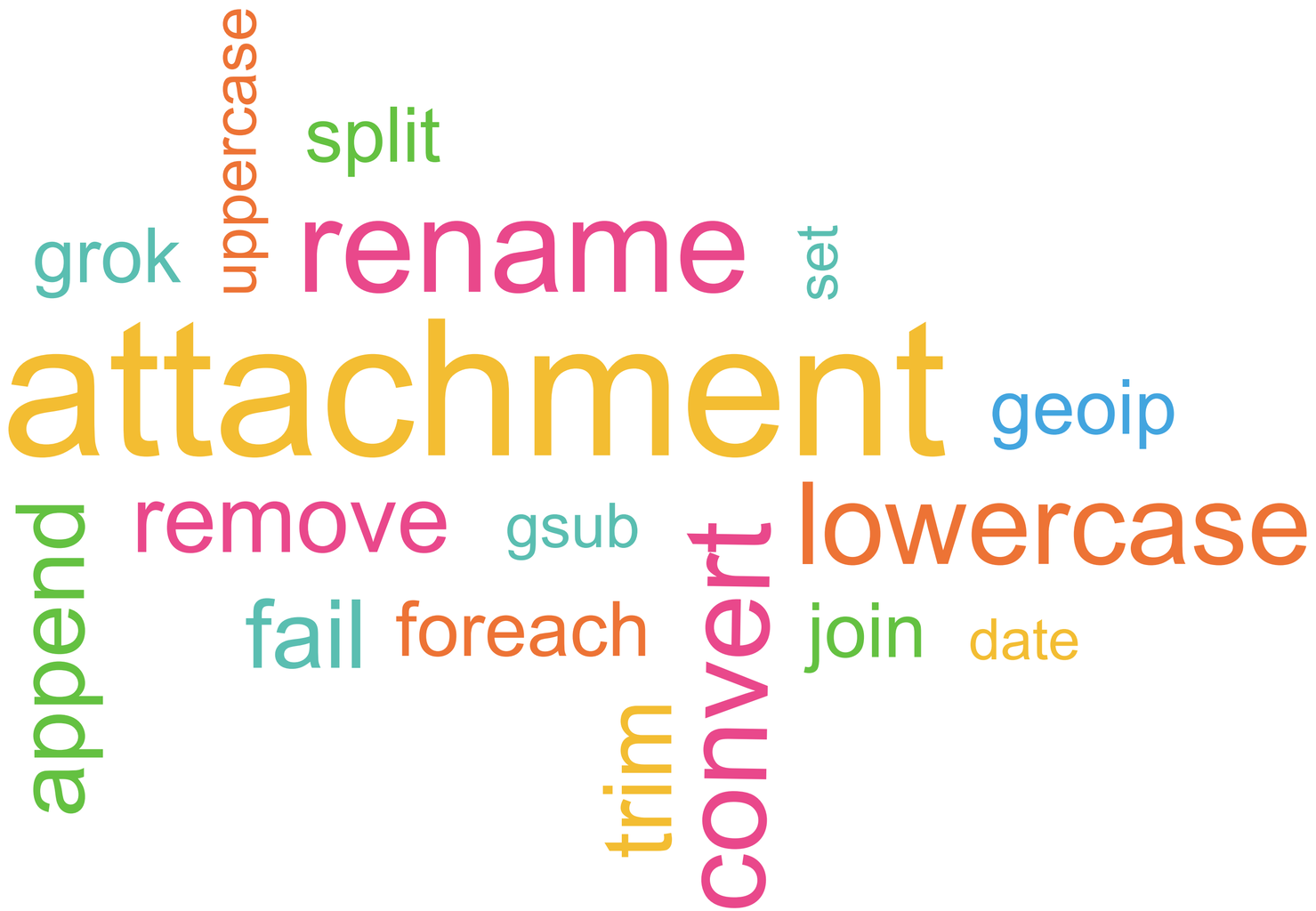
elastic

Ingest pipeline



Pipeline: a set of processors

elastic


uppercase
split
grok
rename
set
attachment
geoip
append
remove
gsub
lowercase
fail
foreach
join
date
trim
convert

Grok processor

Extracts structured fields out of a single text field

```
{
  "grok": {
    "field": "message",
    "patterns": ["%{DATE:date}"]
  }
}
```

Mutate processors

set, remove, rename, convert, gsub, split, join, lowercase, uppercase, trim, append

```
{
  "remove": {
    "field": "message"
  }
}
```

Date processor

Parses a date from a string

```
{
  "date": {
    "field": "timestamp",
    "formats": ["YYYY"]
  }
}
```

Geoip processor

Adds information about the geographical location of IP addresses

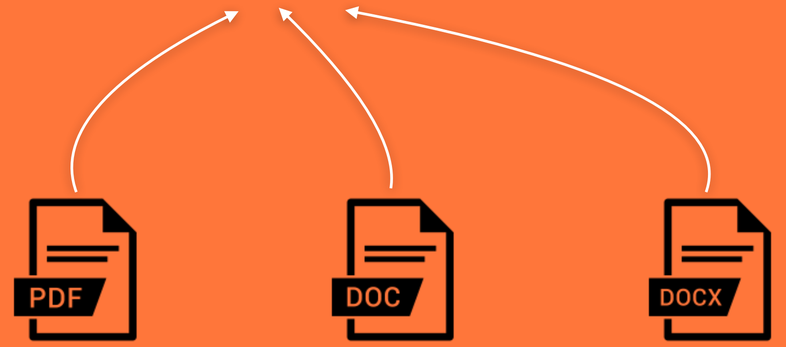
```
{
  "geoip": {
    "field": "ip"
  }
}
```

Attachment processor

You know, for documents

```
{
  "attachment": {
    "field" : "file"
  }
}

// Send a binary content
{
  "file": "BASE64"
}
```



Plugins

Introducing new processors is as easy as writing a plugin

```
{
   "your_plugin": {
      ...
   }
}
```

Pipeline management

```
PUT /_ingest/pipeline/apache-log
{
  "processors" : [
    {
      "grok" : {
        "field": "message",
        "patterns": ["%{COMMONAPACHELOG}"]
      }
    },
    {
      "date" : {
        "field" : "timestamp",
        "formats" : ["dd/MMM/YYYY:HH:mm:ss Z"]
      }
    },
    {
      "remove" : {
        "field" : "message"
      }
    }
  ]
}
```

Where can ingest pipelines be used?

elastic

Index API

```
PUT /apache/doc/1
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / HTTP/1.1\" 200 24"
}
```

Index API

```
PUT /apache/doc/1?pipeline=apache-log
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / HTTP/1.1\" 200 24"
}
```

Bulk API

```
PUT /apache/doc/_bulk
{"index":{}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / ..."}\n
{"index":{}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:07 +0200] \"GET /foo/ ..."}\n
{"index":{}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:09 +2000] \"GET /f.png ..."}\n
```

Bulk API

```
PUT /apache/doc/_bulk?pipeline=apache-log
{"index":{}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / ..."}\n
{"index":{}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:07 +0200] \"GET /foo/ ..."}\n
{"index":{}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:09 +2000] \"GET /f.png ..."}\n
```

Bulk API

```
PUT /_bulk
{"index":{"_index":"apache","_type":"doc"}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / ..."}\n
{"index":{"_index":"mysql","_type":"doc"}}\n
{"message":"..."}\n
```

Bulk API

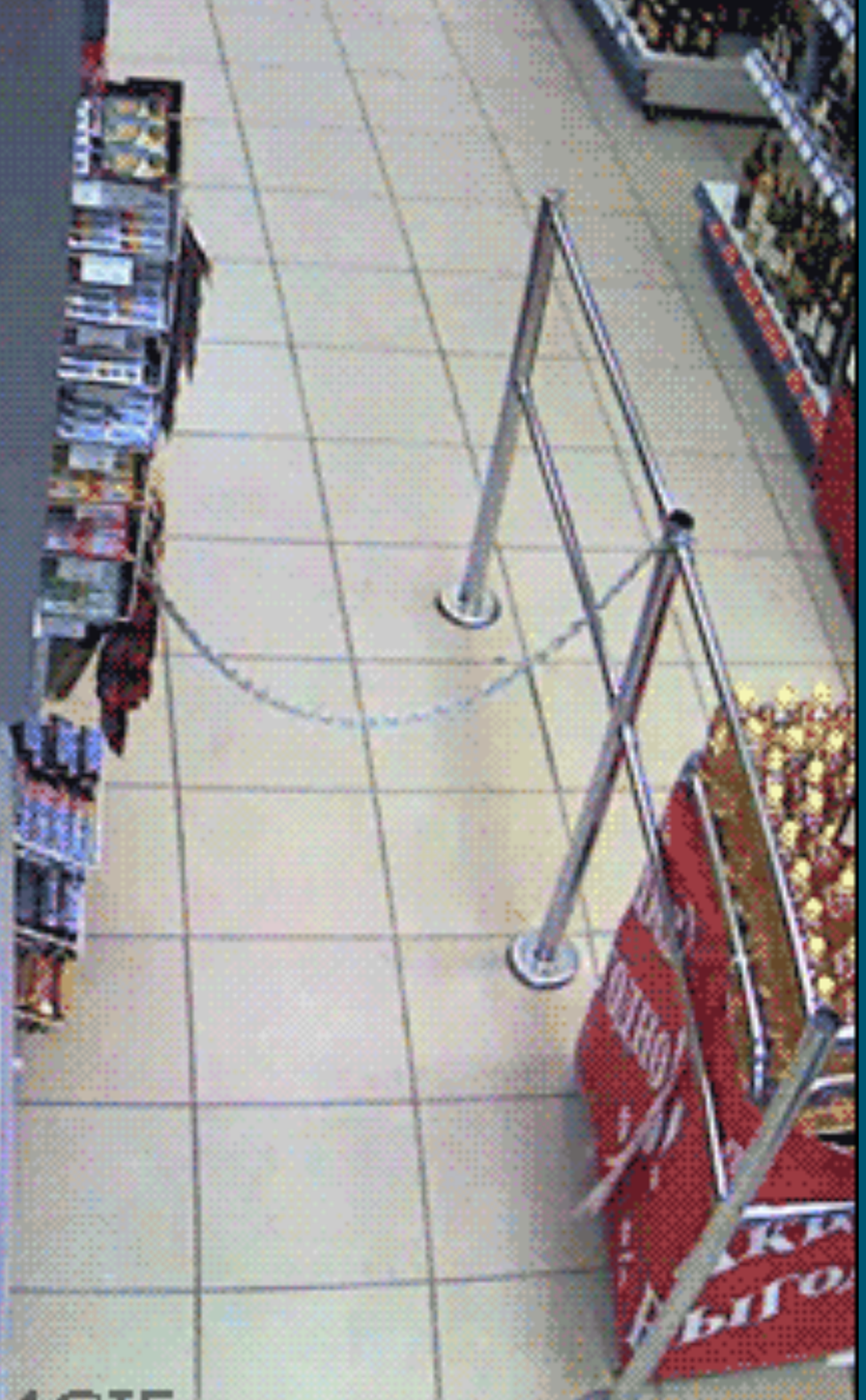
```
PUT /_bulk
{"index":{"_index":"apache","_type":"doc","pipeline":"apache-log"}}\n
{"message":"127.0.0.1 - - [19/Apr/2016:12:00:04 +0200] \"GET / ..."}\n
{"index":{"_index":"mysql","_type":"doc","pipeline":"mysql-log"}}\n
{"message":"..."}\n
```

Reindex API

```
POST /_reindex
{
  "source": {
    "index": "logs",
    "type": "apache"
  },
  "dest": {
    "index": "apache-logs",
    "type": "doc"
  }
}
```

Reindex API

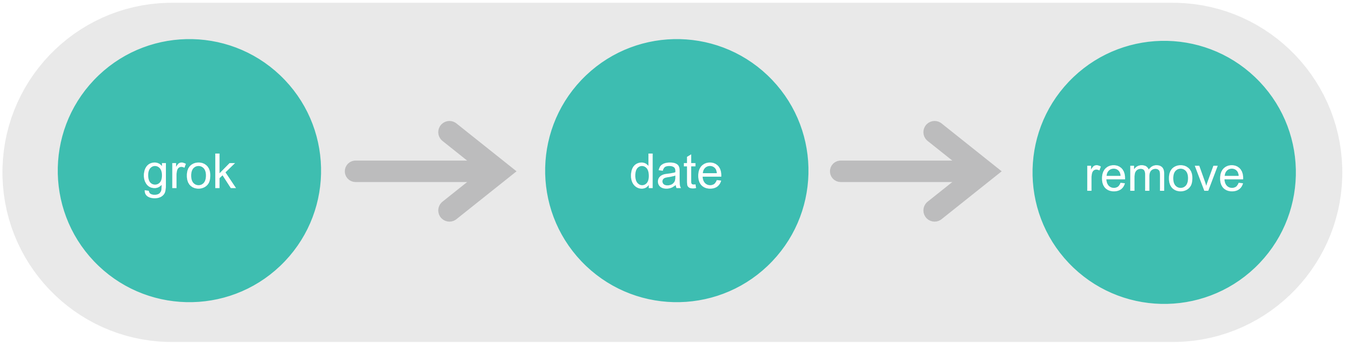
```
POST /_reindex
{
  "source": {
    "index": "logs",
    "type": "apache"
  },
  "dest": {
    "index": "apache-logs",
    "type": "doc",
    "pipeline" : "apache-log"
  }
}
```



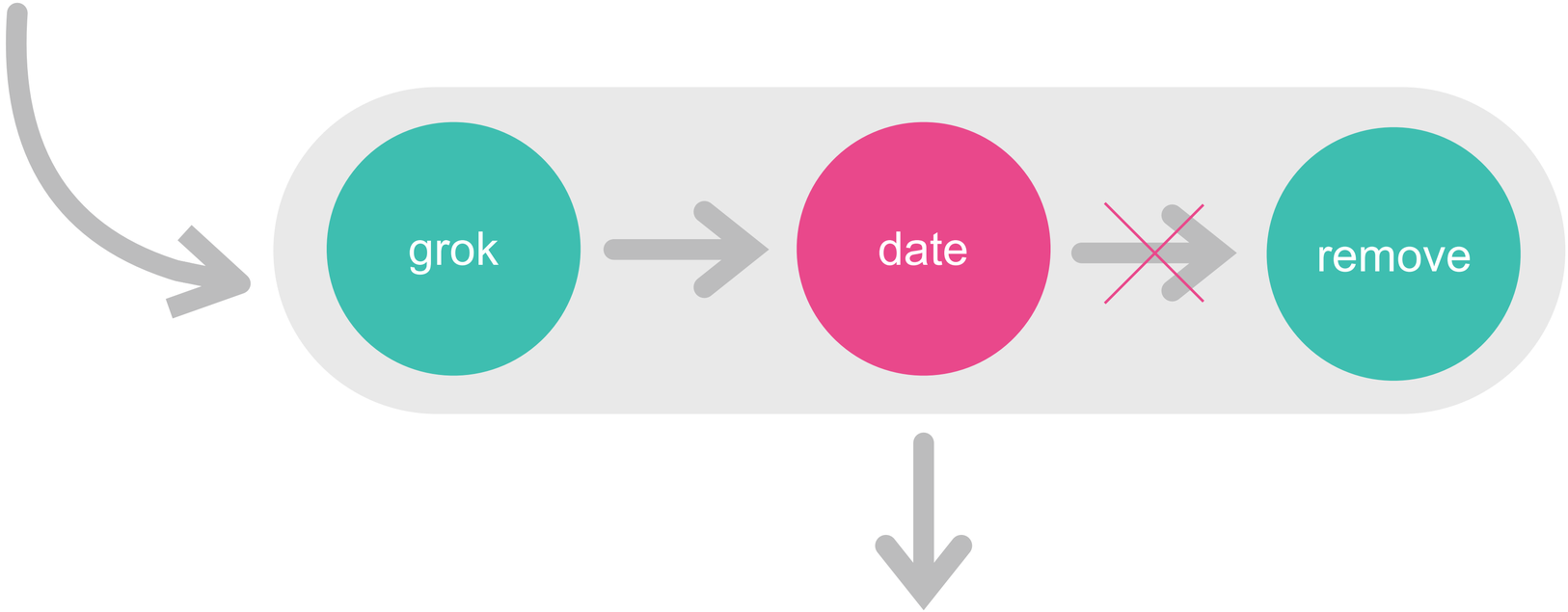
Error handling

elastic

```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \"GET / HTTP/1.1\" 200 24"
}
```



```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \"GET / HTTP/1.1\" 200 24"
}
```

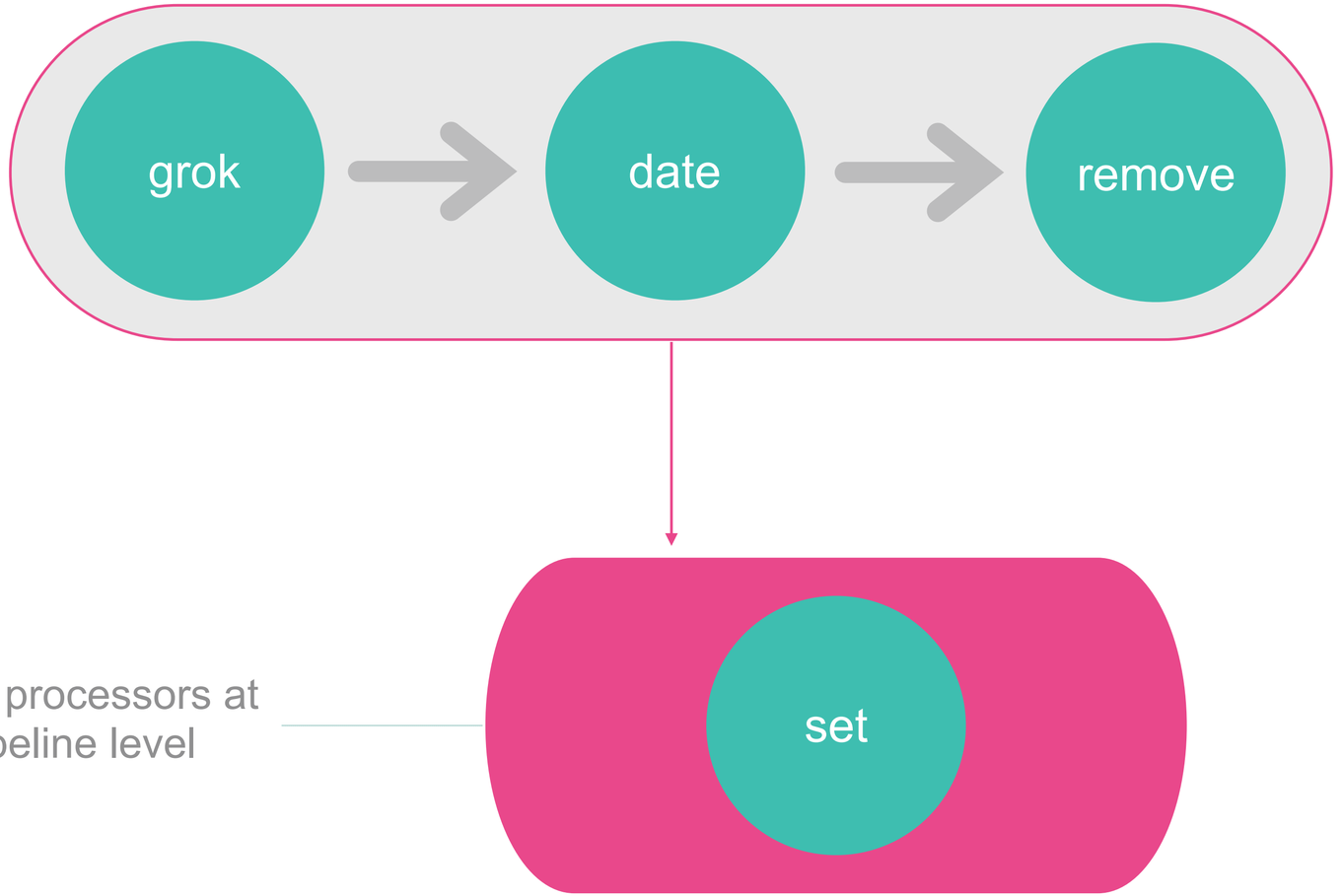


400 Bad Request

unable to parse date [19/Apr/2016:12:00:00 +040]

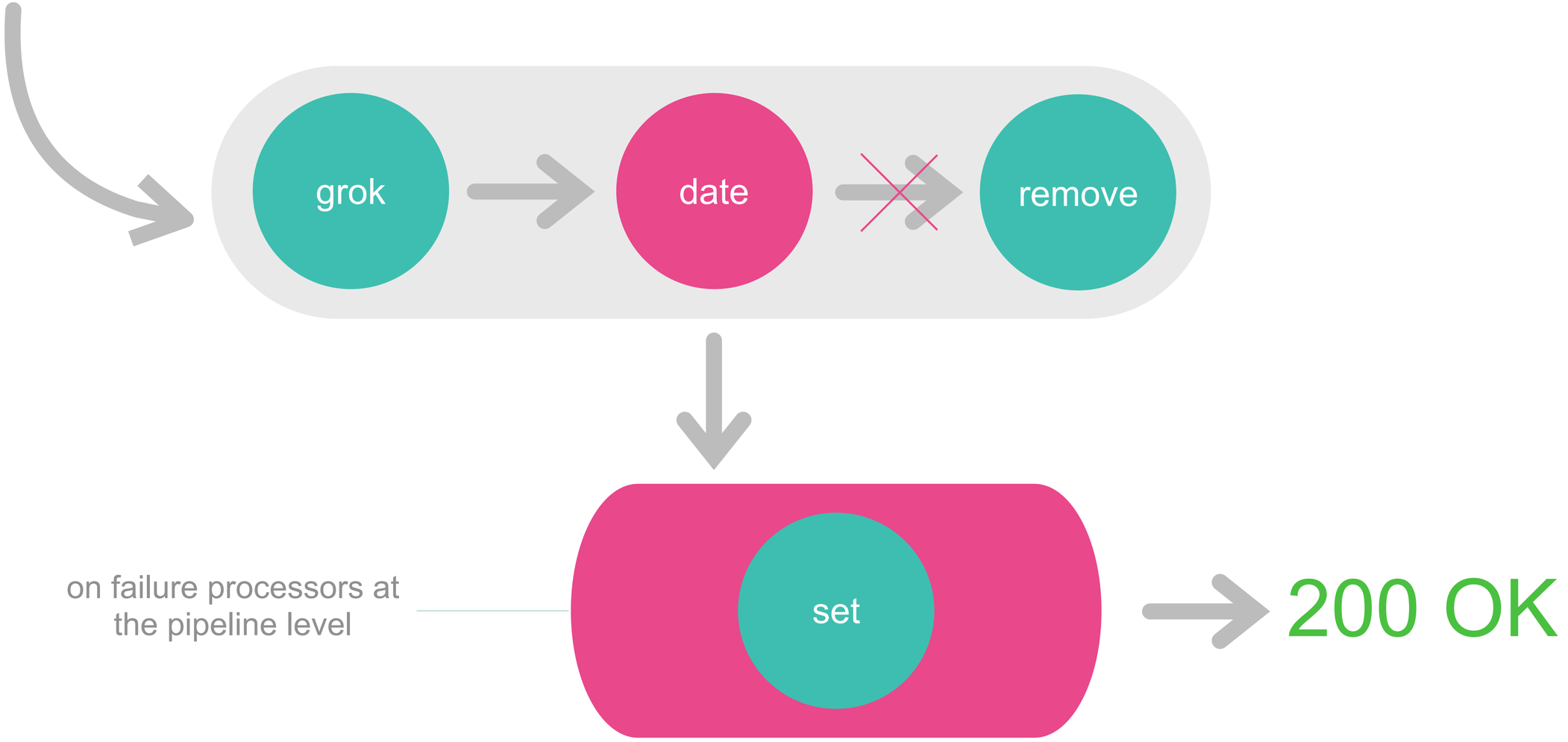
elastic

```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \"GET / HTTP/1.1\" 200 24"
}
```

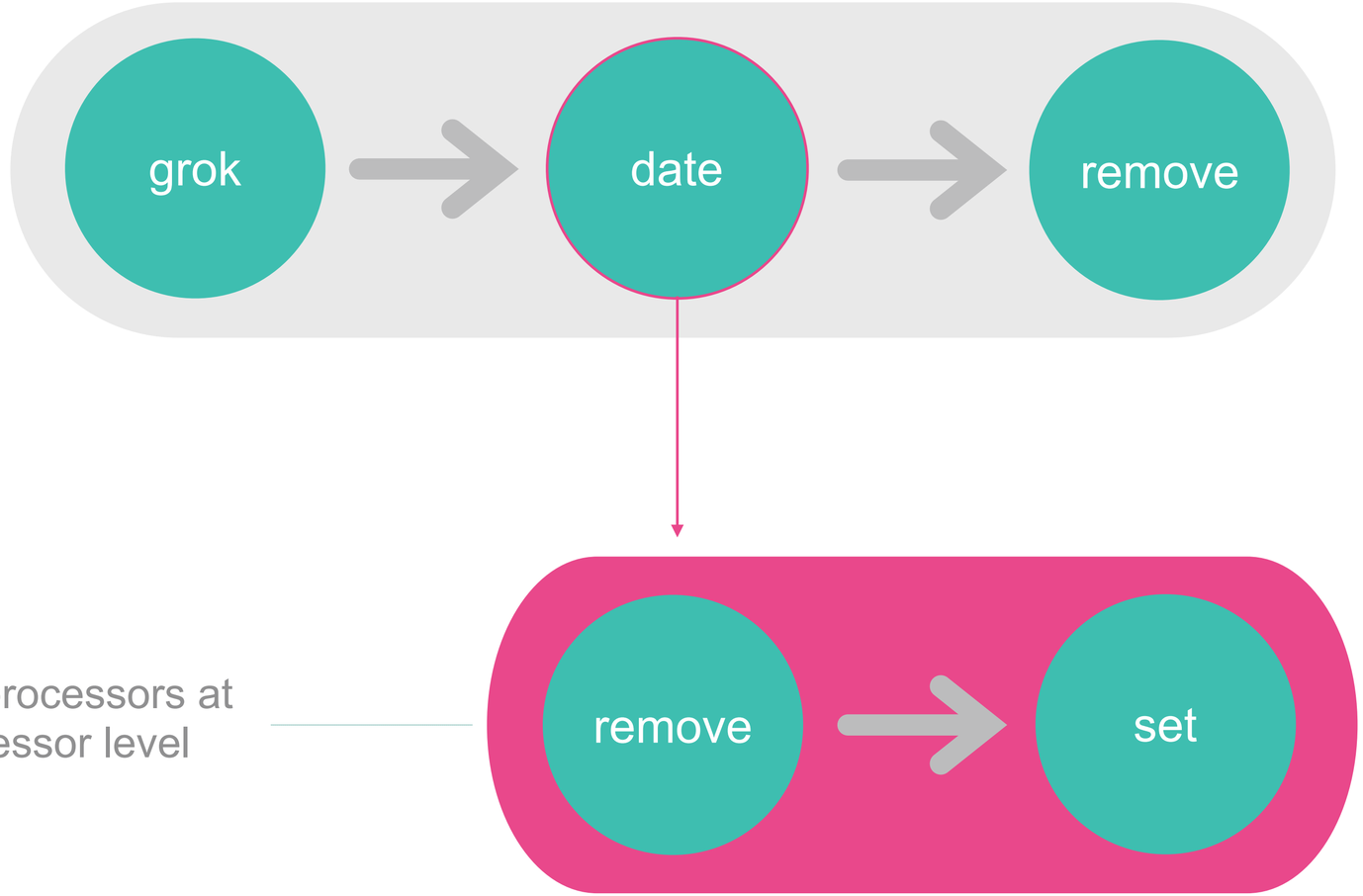


elastic

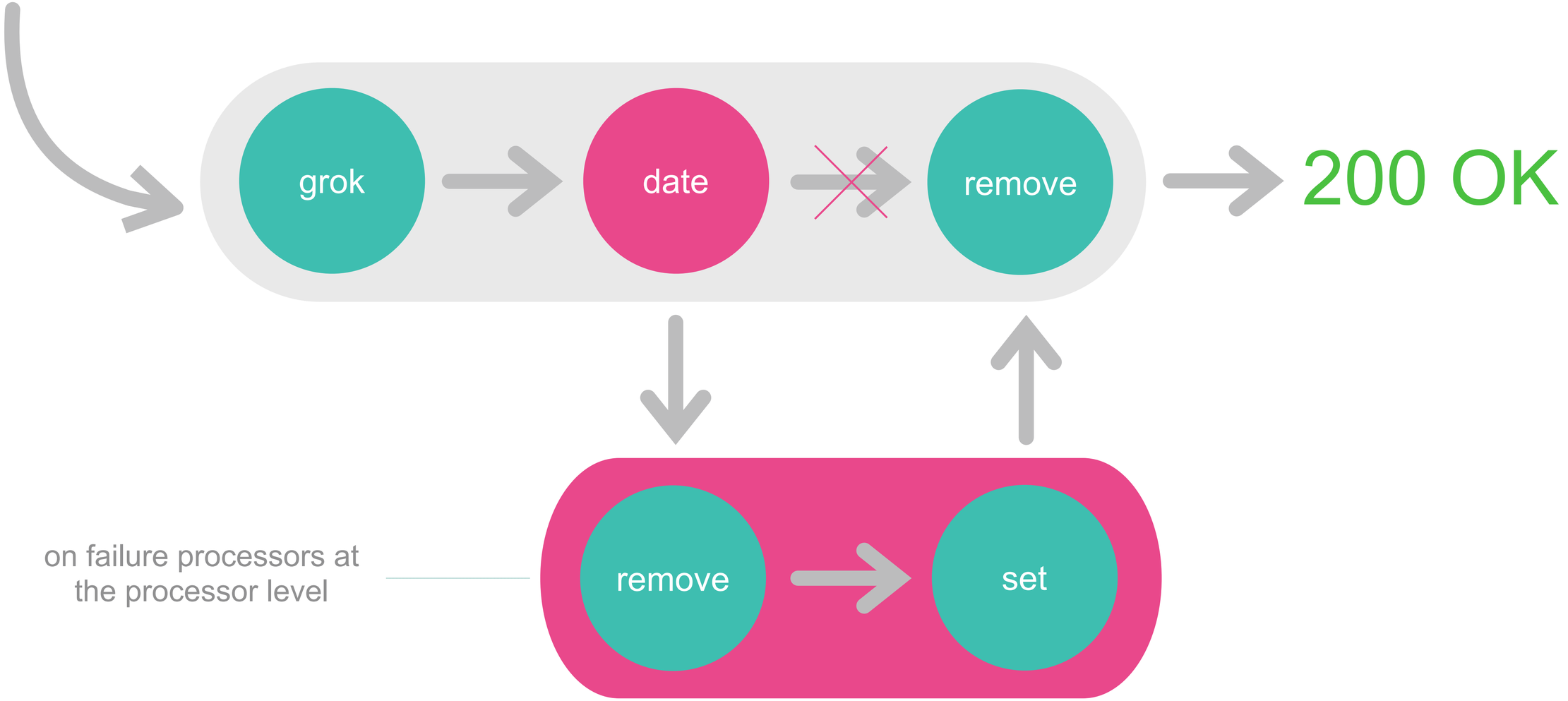
```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \"GET / HTTP/1.1\" 200 24"
}
```

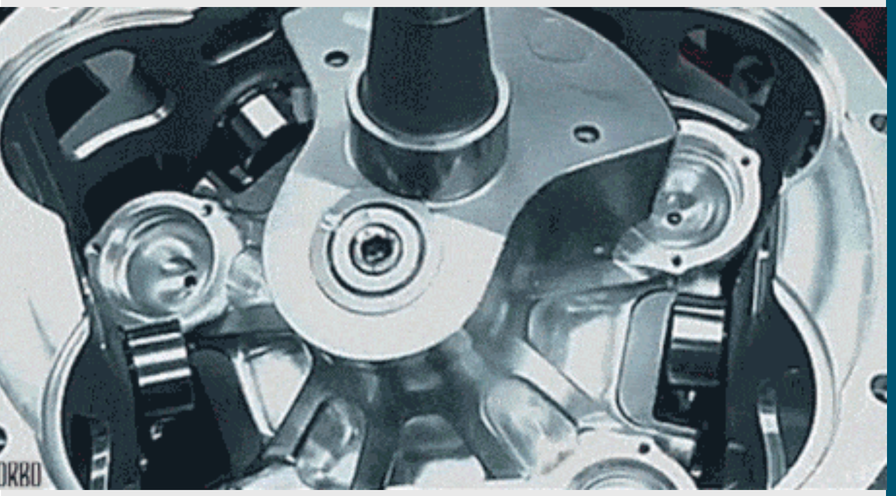


```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \"GET / HTTP/1.1\" 200 24"
}
```



```
{
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \"GET / HTTP/1.1\" 200 24"
}
```



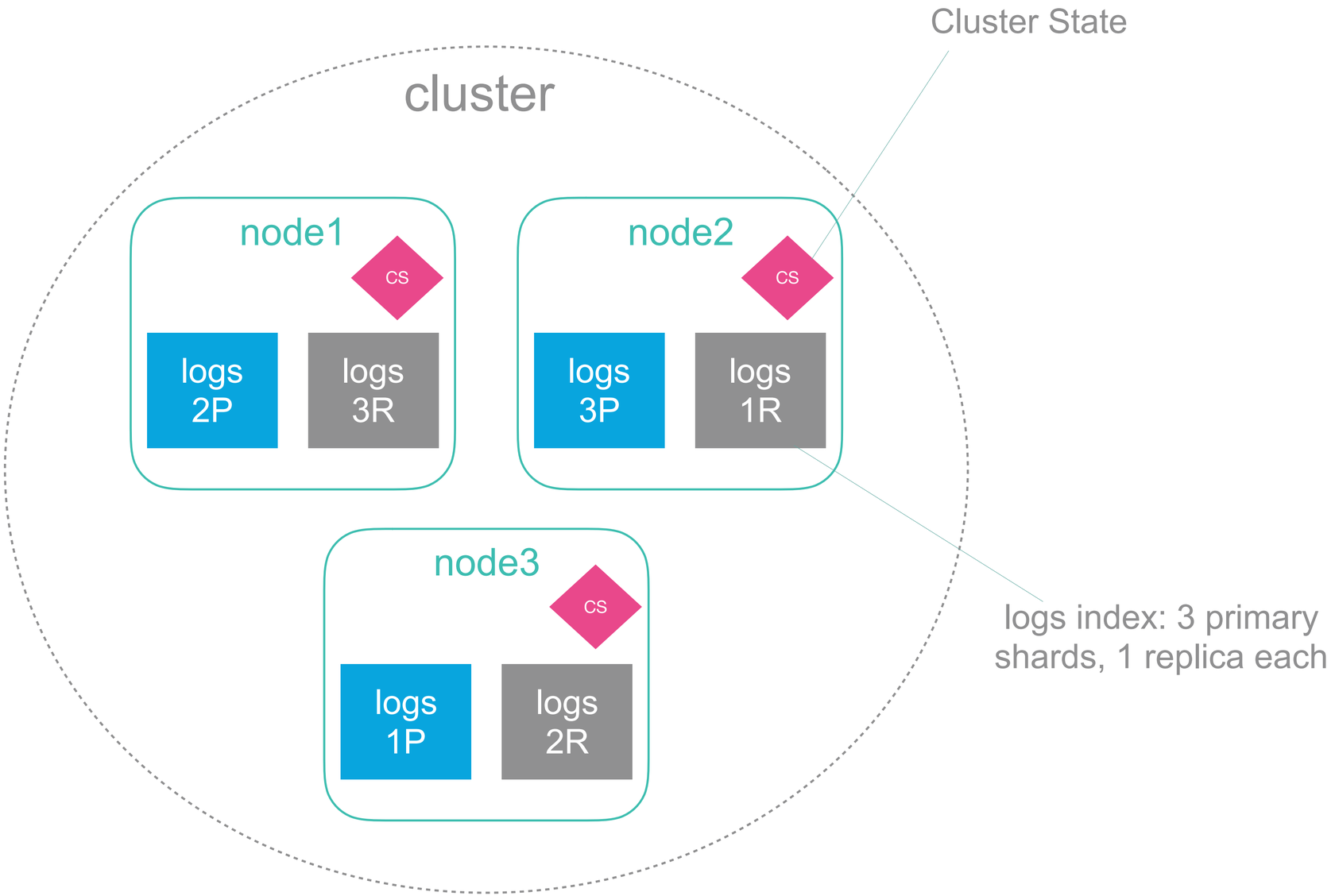


Ingest node internals

elastic

Default scenario

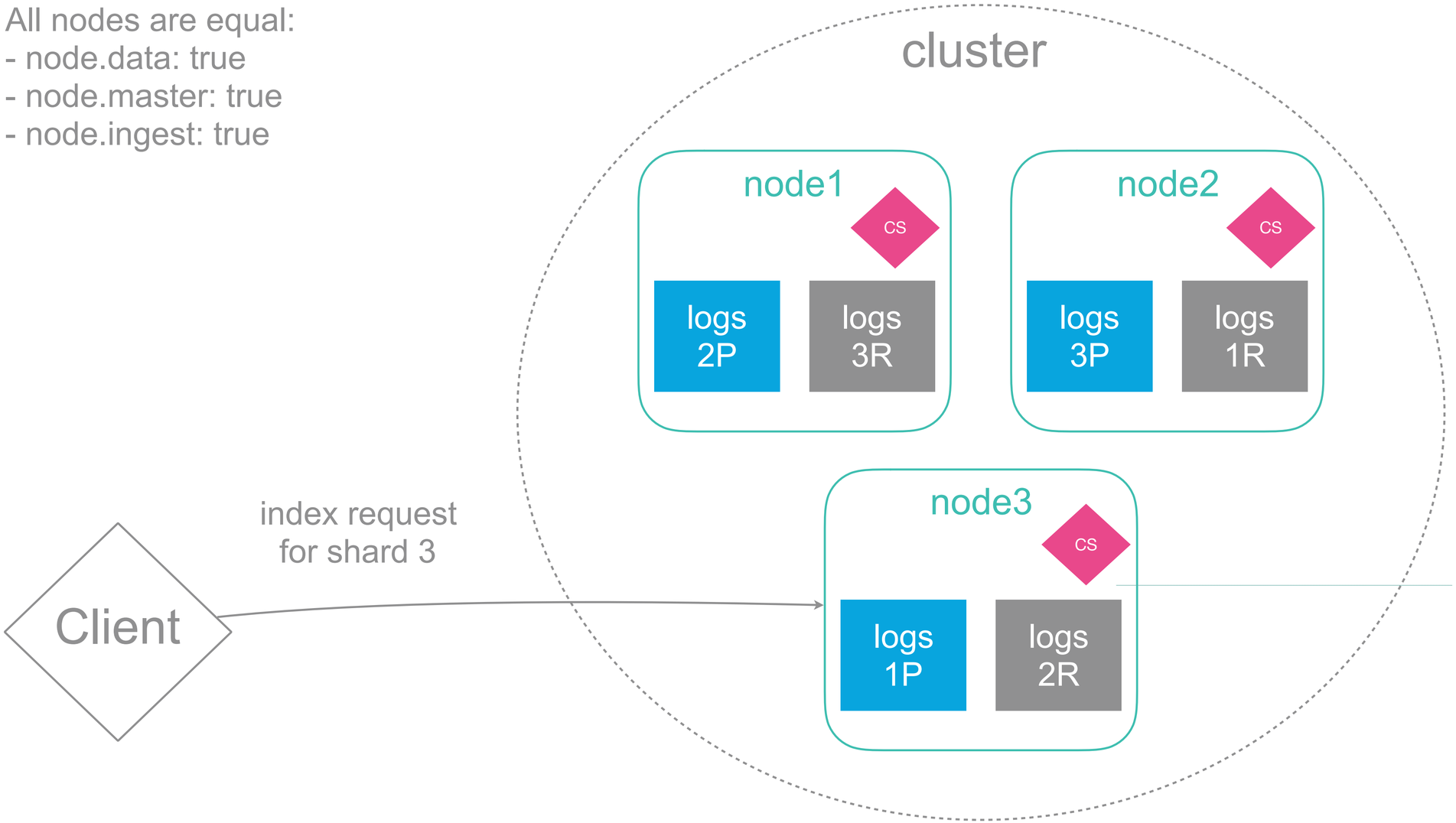
All nodes are equal:
- node.data: true
- node.master: true
- node.ingest: true



Client

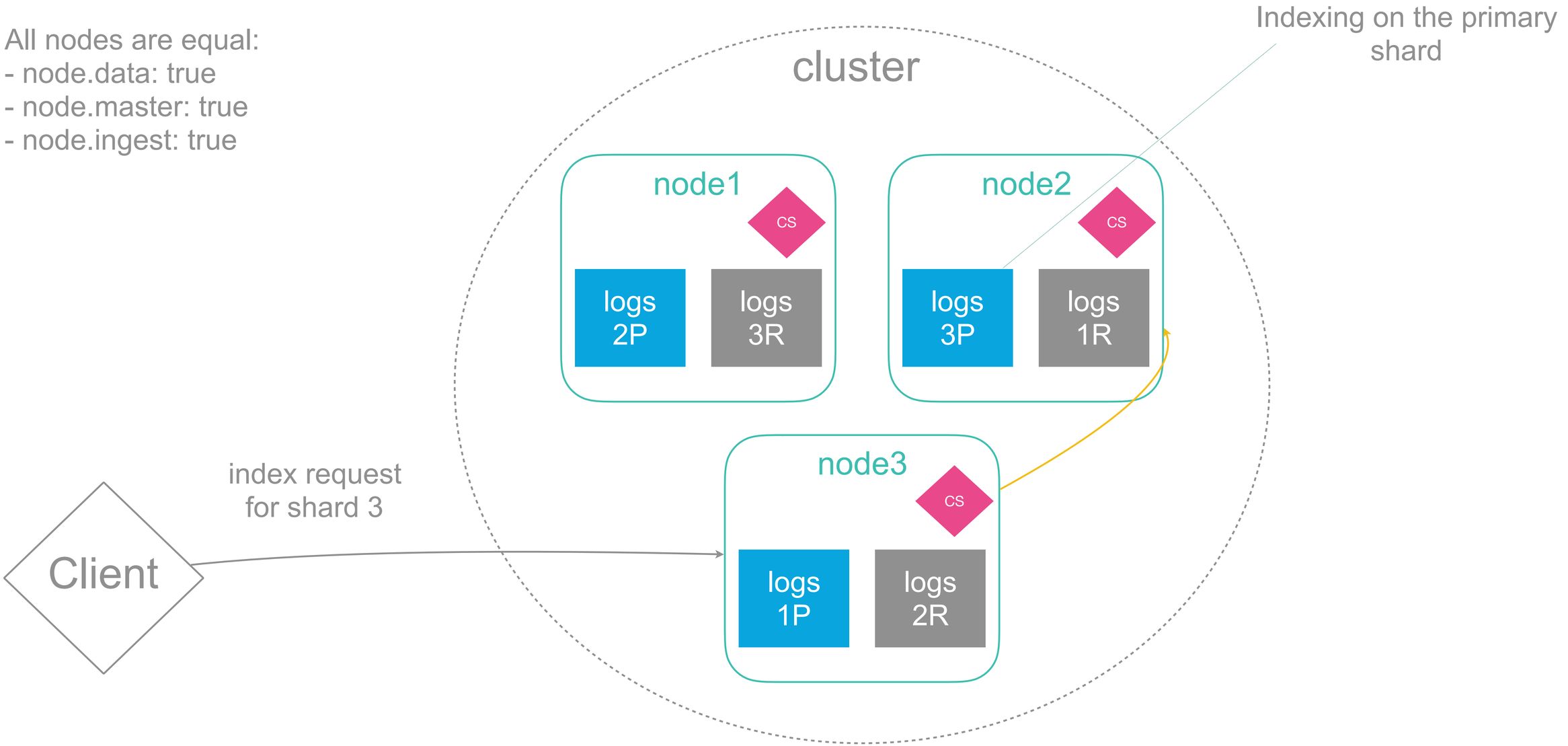
Default scenario

All nodes are equal:
- node.data: true
- node.master: true
- node.ingest: true



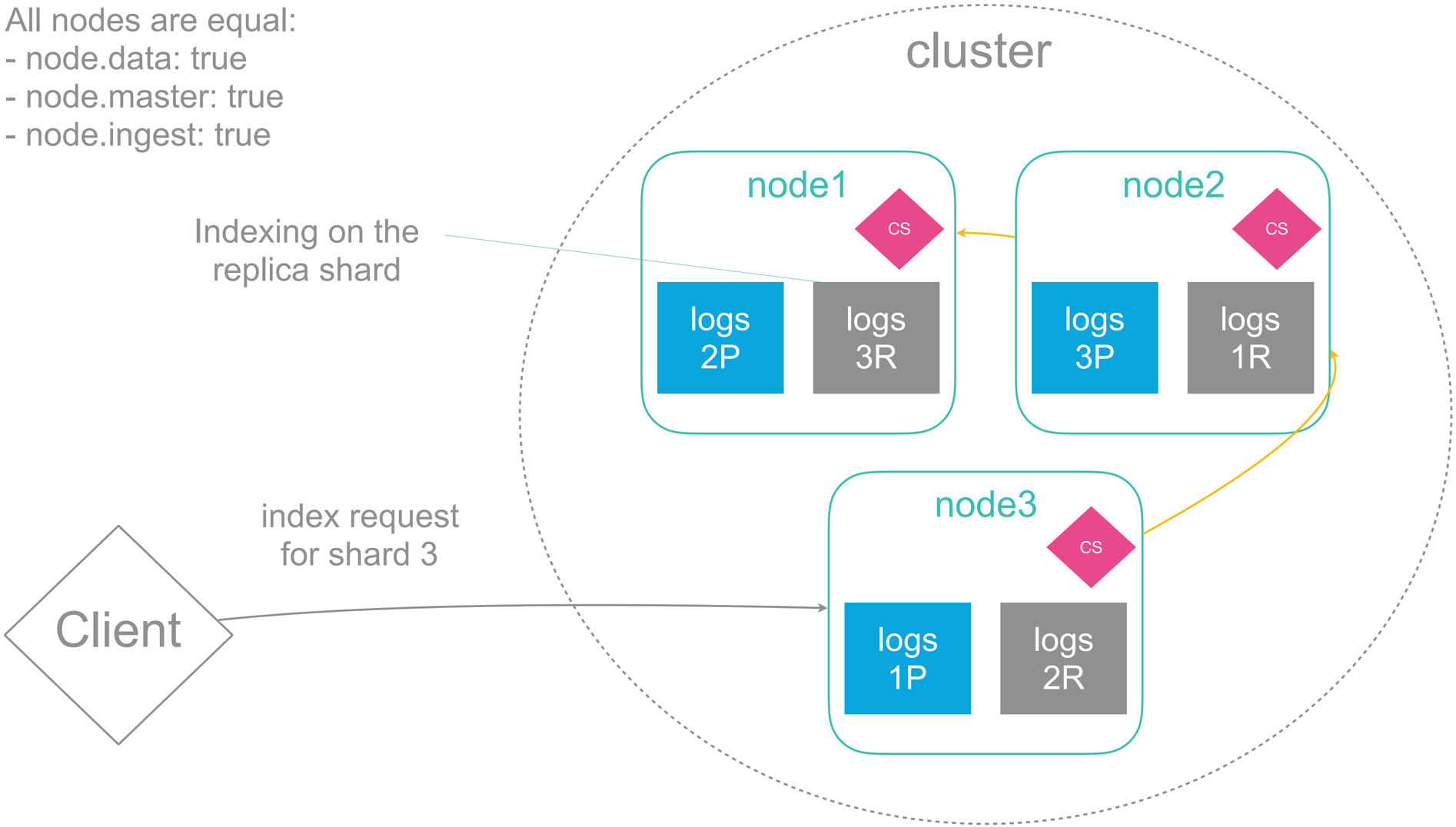
Default scenario

All nodes are equal:
- node.data: true
- node.master: true
- node.ingest: true



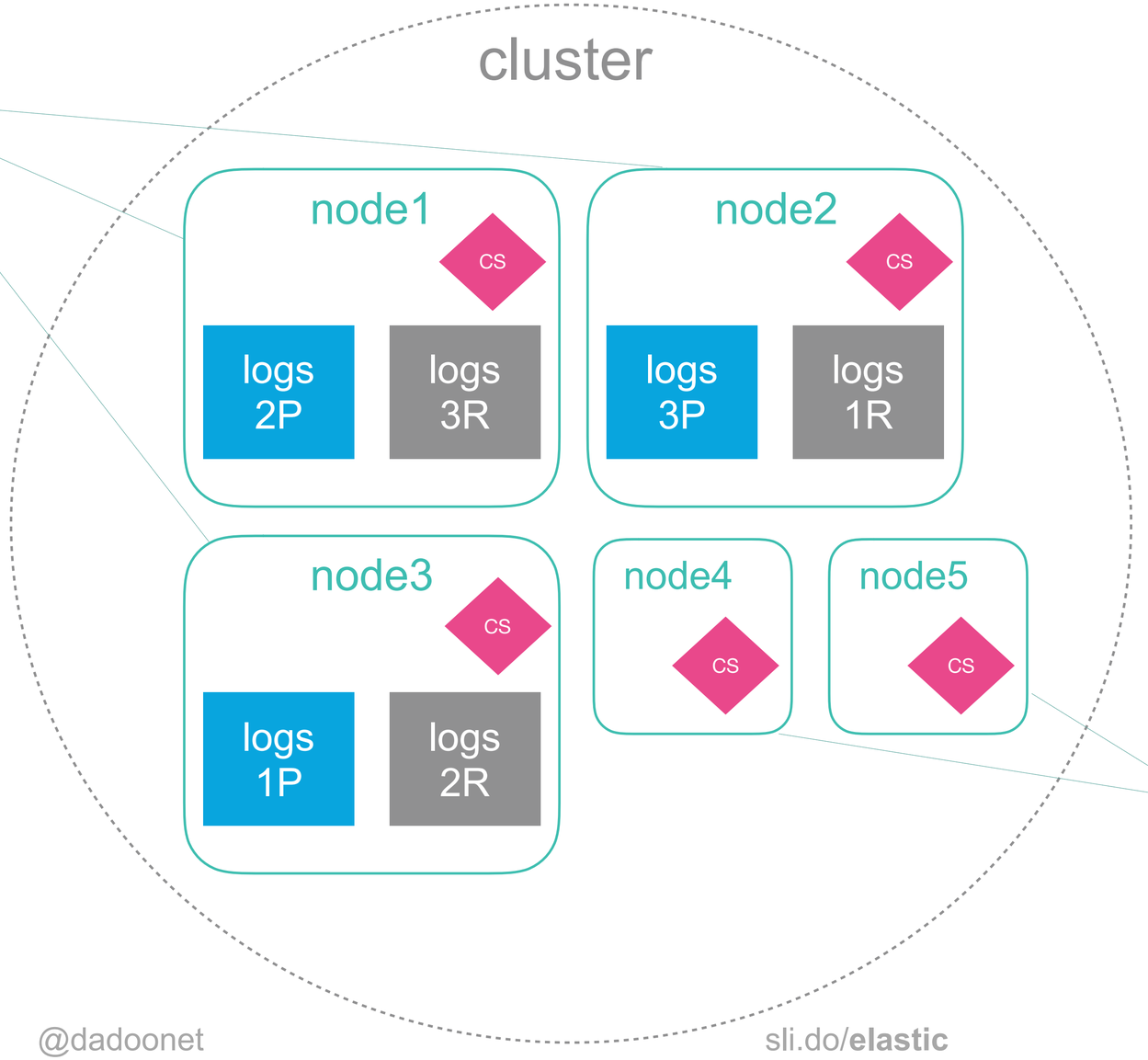
Default scenario

All nodes are equal:
- node.data: true
- node.master: true
- node.ingest: true



Ingest dedicated nodes

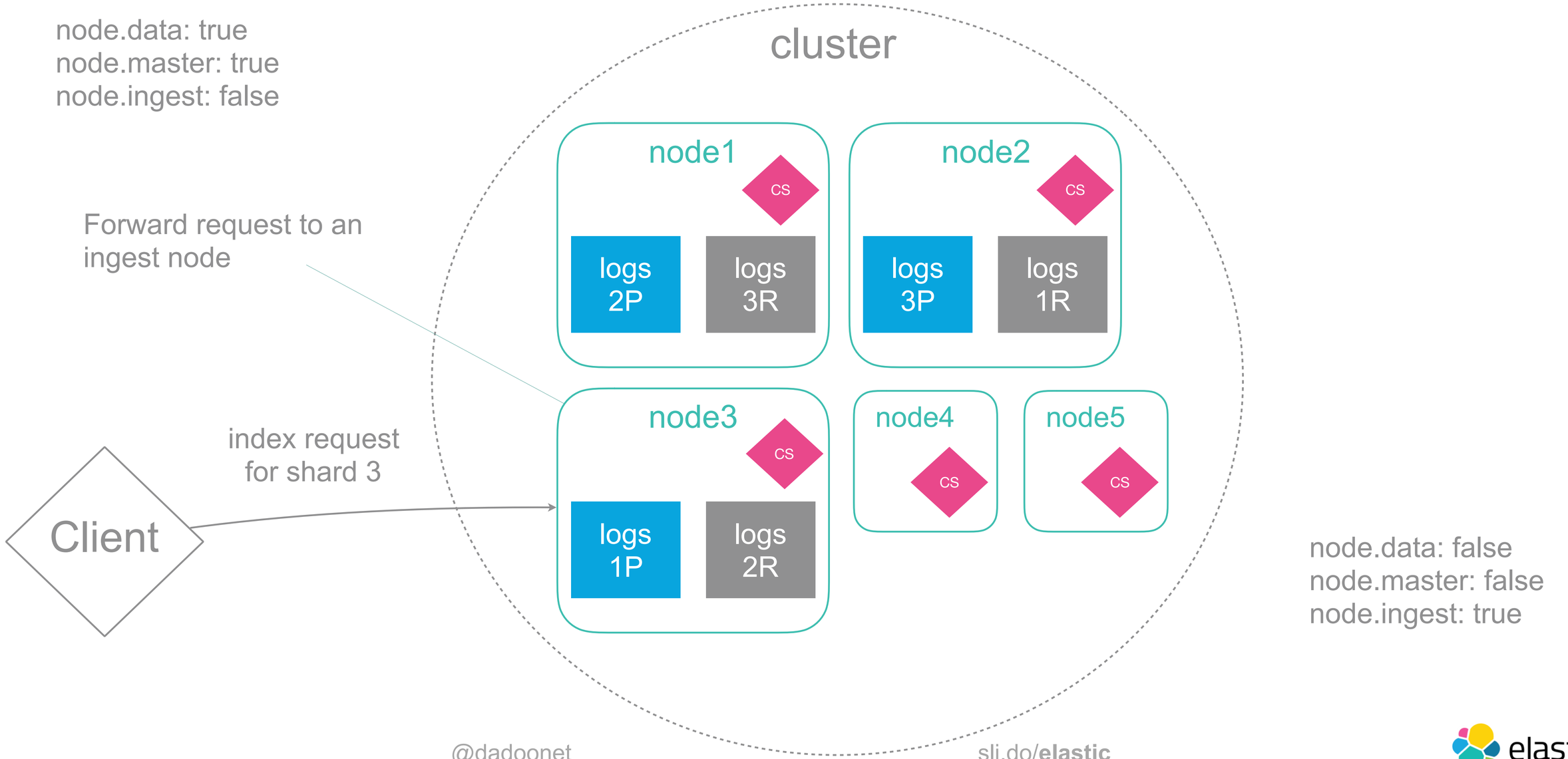
node.data: true
node.master: true
node.ingest: false



Client

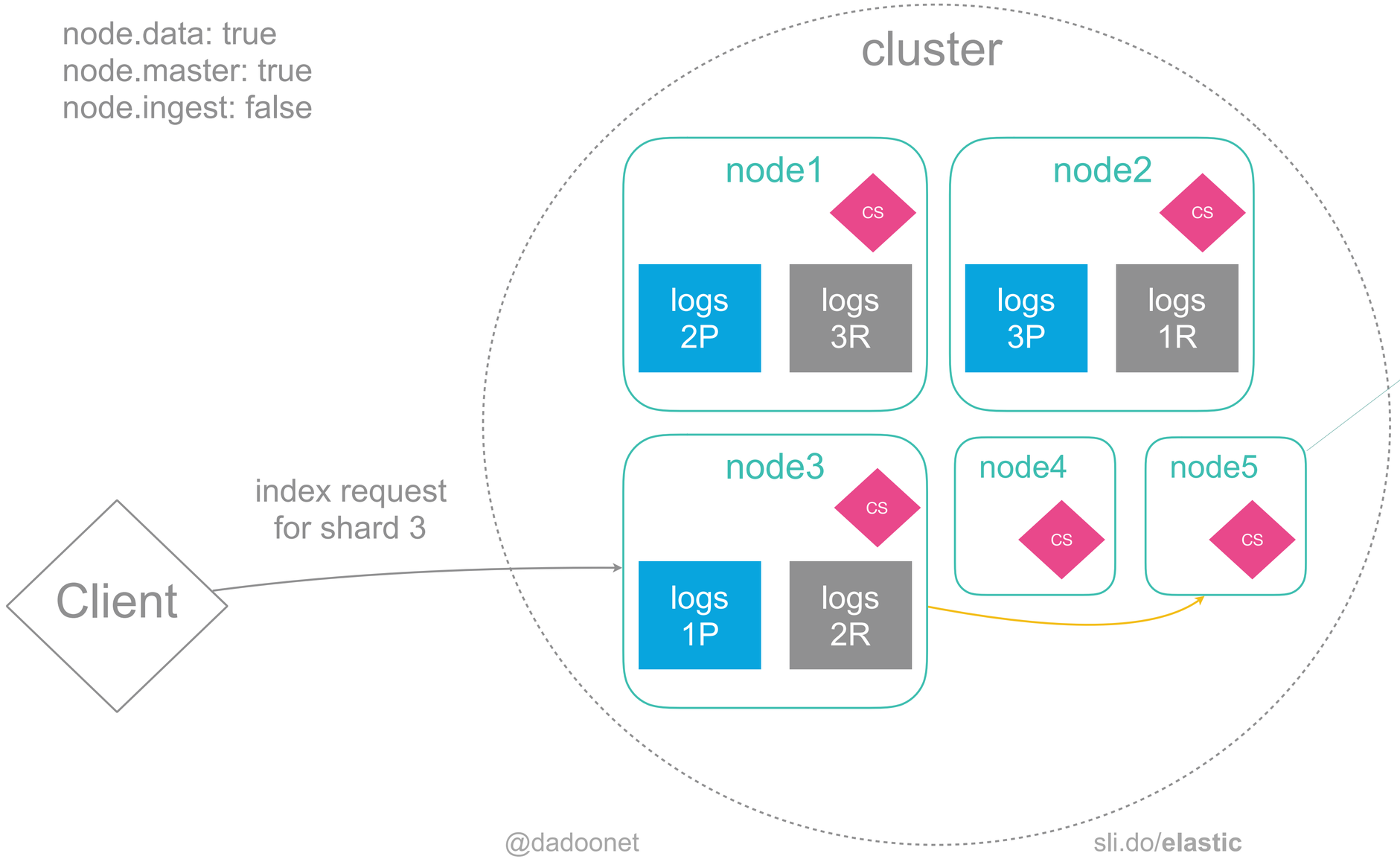
node.data: false
node.master: false
node.ingest: true

Ingest dedicated nodes



Ingest dedicated nodes

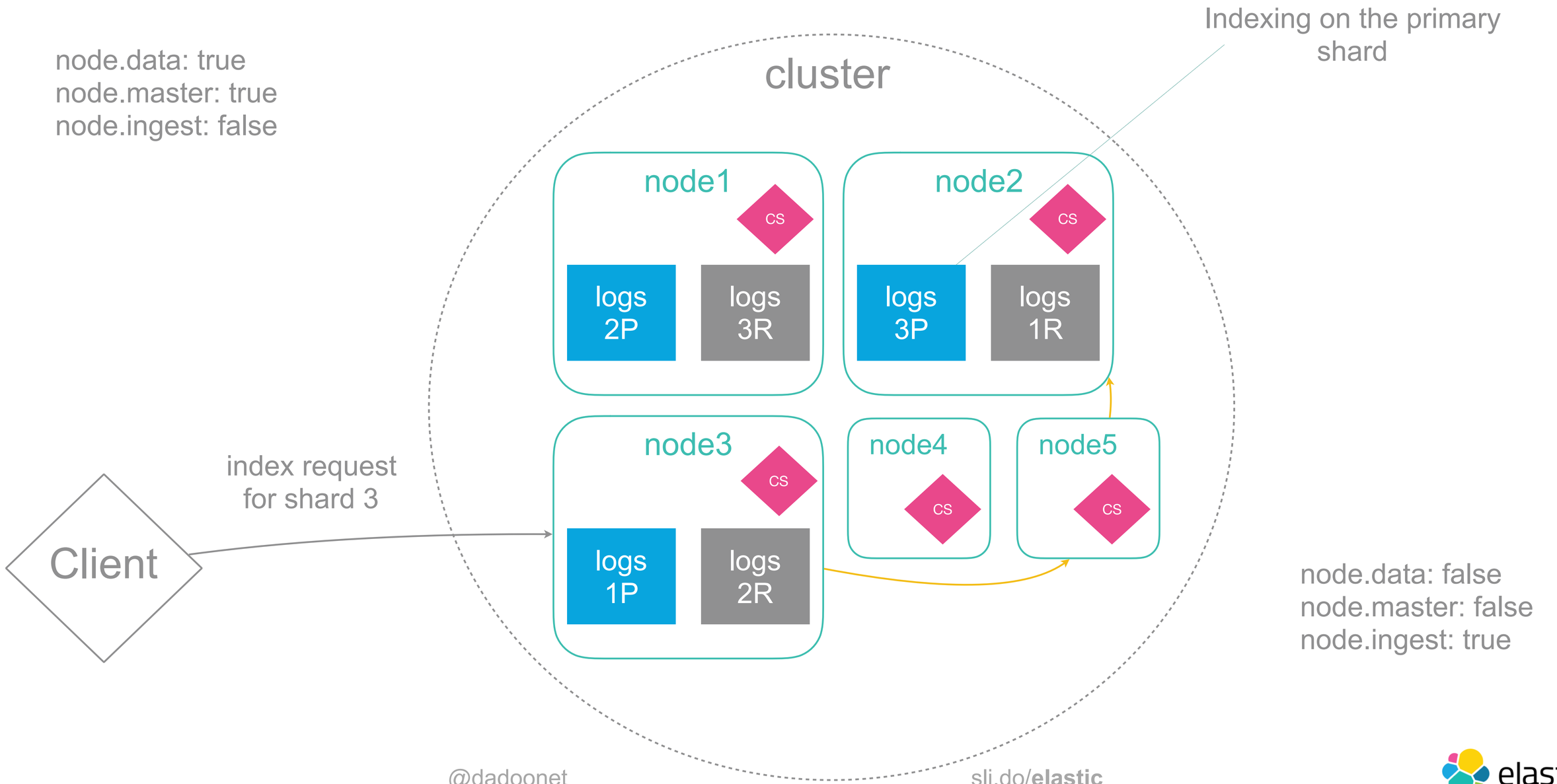
node.data: true
node.master: true
node.ingest: false



node.data: false
node.master: false
node.ingest: true

elastic

Ingest dedicated nodes



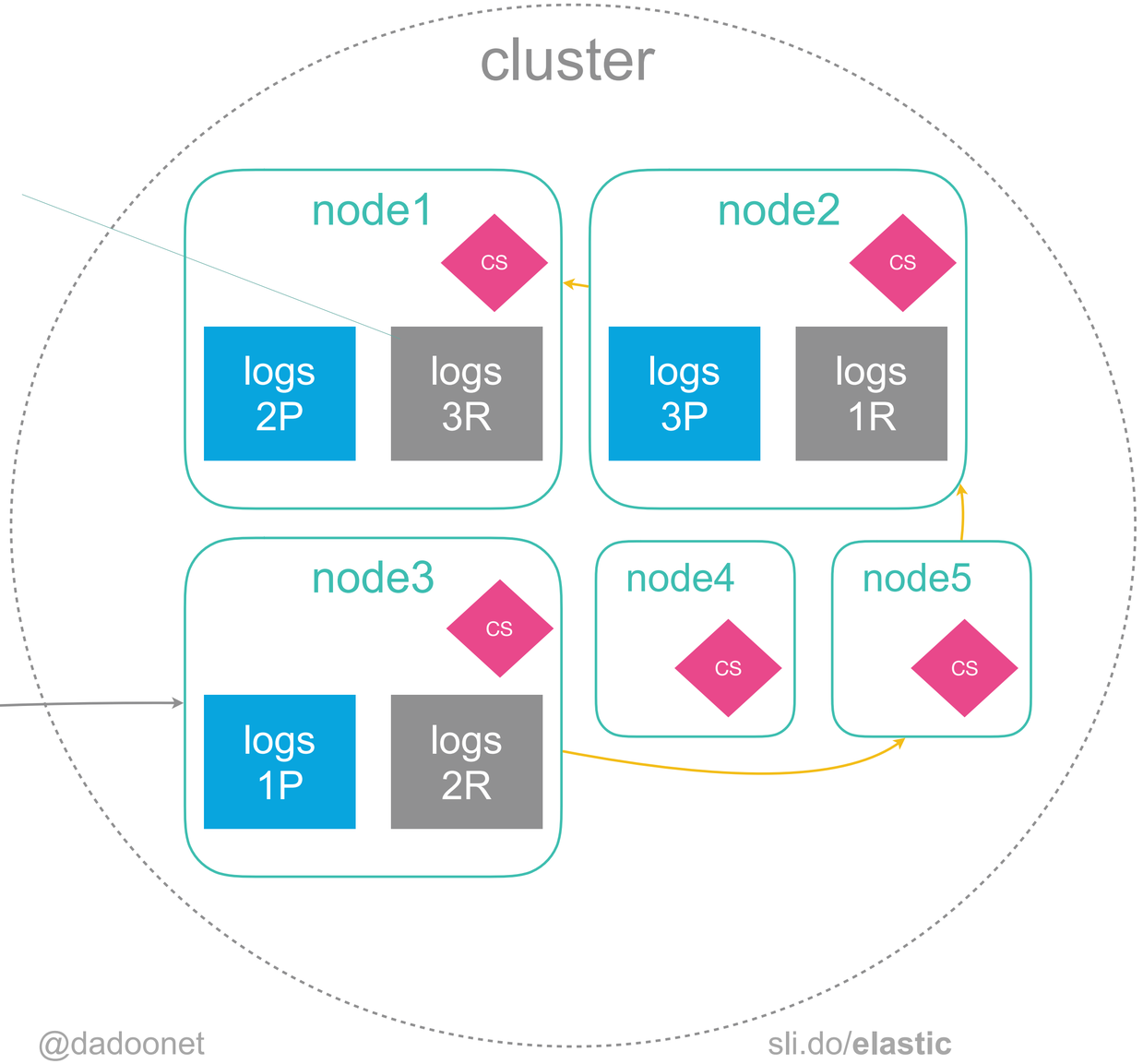
Ingest dedicated nodes

node.data: true
node.master: true
node.ingest: false

Indexing on the replica shard

index request for shard 3

Client



node.data: false
node.master: false
node.ingest: true

Demo time!

```
52.35.38.35 - - [19/Apr/2016:12:00:04 +0200] "GET / HTTP/1.1" 200 24
```

elastic

bano-ingest plugin

From postal address to geo_point
From geo_point to postal address

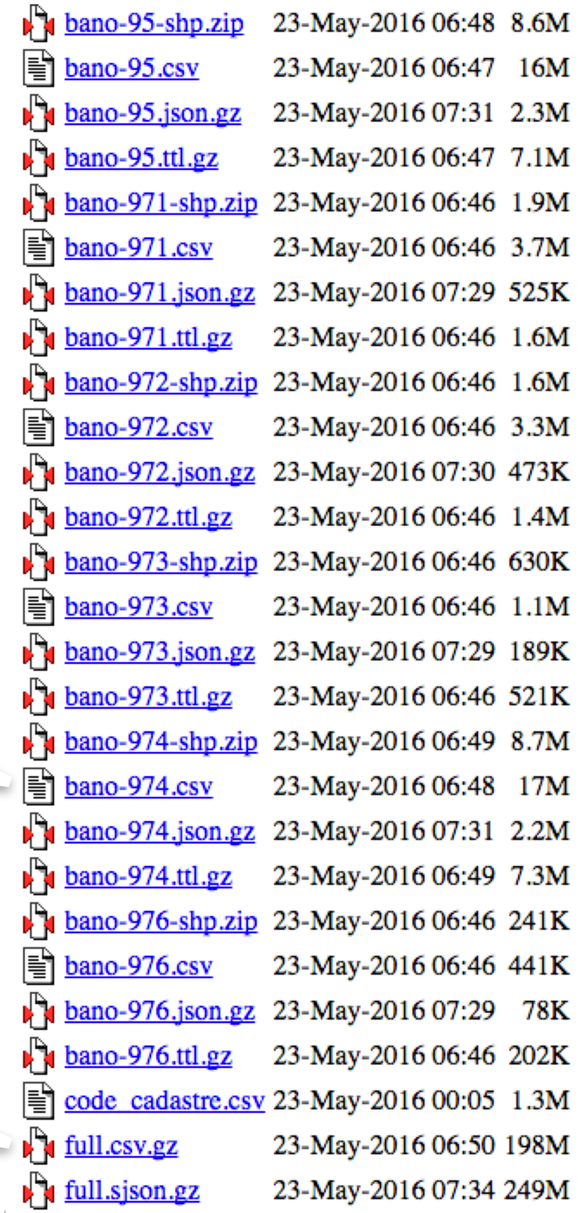
elastic

What is BANO?

- French Open Data base for postal addresses
- http://openstreetmap.fr/bano
- http://bano.openstreetmap.fr/data/

per region

all addresses



elastic

BANO Format

```
976030950H-26,26,RUE DISMA,97660,Bandrélé,CAD,-12.891701,45.202652
976030950H-28,28,RUE DISMA,97660,Bandrélé,CAD,-12.891900,45.202700
976030950H-30,30,RUE DISMA,97660,Bandrélé,CAD,-12.891781,45.202535
976030950H-32,32,RUE DISMA,97660,Bandrélé,CAD,-12.892005,45.202564
976030950H-3,3,RUE DISMA,97660,Bandrélé,CAD,-12.892444,45.202135
976030950H-34,34,RUE DISMA,97660,Bandrélé,CAD,-12.892068,45.202450
976030950H-4,4,RUE DISMA,97660,Bandrélé,CAD,-12.892446,45.202367
976030950H-5,5,RUE DISMA,97660,Bandrélé,CAD,-12.892461,45.202248
976030950H-6,6,RUE DISMA,97660,Bandrélé,CAD,-12.892383,45.202456
976030950H-8,8,RUE DISMA,97660,Bandrélé,CAD,-12.892300,45.202555
976030950H-9,9,RUE DISMA,97660,Bandrélé,CAD,-12.892355,45.202387
976030951J-103,103,RTE NATIONALE 3,97660,Bandrélé,CAD,-12.893639,45.201696
  \_ ID          |   \_ Street Name |          \     \_ Source  \_ Geo point
                 |                  |           \
                 |_ Street Number   |_ Zipcode   \_ City Name
```

Import bano dataset

elastic

Load CSV with Logstash (Extract: input)

```
976030951J-103,103,RTE NATIONALE 3,97660,Bandrélé,CAD,-12.893639,45.201696
  \_ ID          |    \_ Street Name |          \      \_ Source  \_ Geo point
                 |                   |           \
                 |_ Street Number    |_ Zipcode   \_ City Name
```

```
input {
  stdin { }
}
```

```
{
  "message":"976030951J-103,103,RTE NATIONALE 3,97660,Bandrélé,CAD,-12.893639,45.201696",
  "@timestamp":"2017-12-05T16:00:00.000PST",
  "@version":1,
  "host":"MacBook-Pro-David.local"
}
```

Load CSV with Logstash (Transform: filter)

```
{
  "message":"976030951J-103,103,RTE NATIONALE 3,97660,Bandrélé,CAD,-12.893639,45.201696",
  "@timestamp":"2017-12-05T16:00:00.000PST",
  "@version":1, "host":"MacBook-Pro-David.local"
}
```

```
filter {
  csv {
    separator => ","
    columns => [
      "id","number","street_name","zipcode","city","source","latitude","longitude"
    ]
    remove_field => [ "message", "@version", "@timestamp", "host" ]
  }
}
```

```
{
  "source":"CAD", "id":"976030951J-103",
  "number":"103", "street_name":"RTE NATIONALE 3",
  "zipcode":"97660", "city":"Bandrélé",
  "latitude":"-12.893639", "longitude":"45.201696"
}
```

Load CSV with Logstash (Transform: filter)

```
filter {
  mutate {
    convert => { "longitude" => "float" }
    convert => { "latitude" => "float" }
    rename => {
      "longitude" => "[location][lon]"
      "latitude" => "[location][lat]"
      "number" => "[address][number]"
      "street_name" => "[address][street_name]"
      "zipcode" => "[address][zipcode]"
      "city" => "[address][city]"
    }
    replace => {
      "region" => "${REGION}"
    }
  }
}
```


```
{
  "source":"CAD","id":"976030951J-103",
  "number":"103",
  "street_name":"RTE NATIONALE 3",
  "zipcode":"97660","city":"Bandrélé",
  "latitude":"-12.893639",
  "longitude":"45.201696"
}
```



```
{
  "source":"CAD","id":"976030951J-103",
  "region":"976",
  "address":{
    "number":"103",
    "street_name":"RTE NATIONALE 3",
    "zipcode":"97660",
    "city":"Bandrélé"
  },
  "location":{
    "lat":-12.893639,"lon":45.201696
  }
}
```


elastic

Load CSV with Logstash (Load: output)

```
{
  "source":"CAD","id":"976030951J-103",
  "region":"976",
  "address":{
    "number":"103","street_name":"RTE NATIONALE 3",
    "zipcode":"97660","city":"Bandrélé"
  },
  "location":{
    "lat":-12.893639,"lon":45.201696
  }
}
```

```
output {
  elasticsearch {
    "template_name" => "bano"
    "template_overwrite" => true
    "template" => "${SOURCE_DIR}/src/main/logstash/bano.json"
    "index" => ".bano-${REGION}"
    "document_id" => "%{[id]}"
  }
}
```

Index template (index settings)

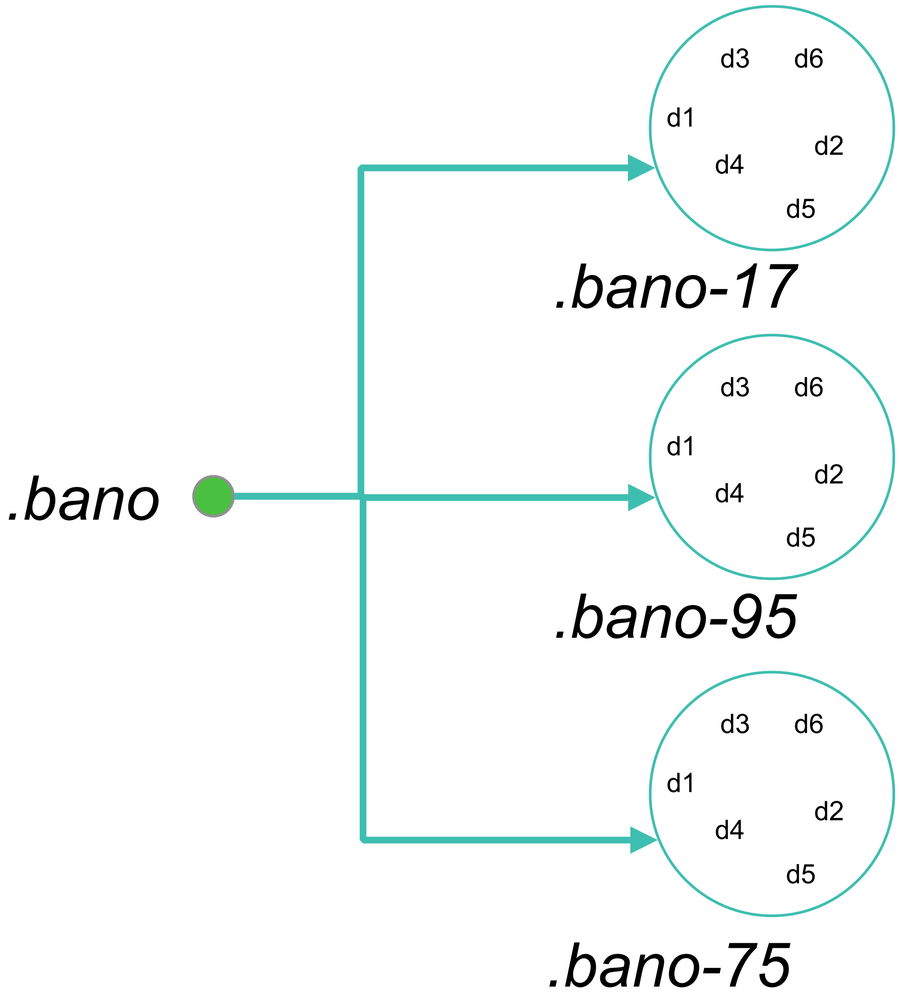
```
{
  "template": ".bano-*", "settings": {
    "index.number_of_shards": 1, "index.number_of_replicas": 0,
    "index.analysis: {
      "analyzer": {
        "bano_analyzer": {
          "type": "custom", "tokenizer": "standard",
          "filter" : [ "lowercase", "asciifolding" ]
        },
        "bano_street_analyzer": {
          "type": "custom", "tokenizer": "standard",
          "filter" : [ "lowercase", "asciifolding", "bano_synonym" ]
        }
      },
      "filter": {
        "bano_synonym": {
          "type": "synonym",
          "synonyms" : [ "bd => boulevard", "av => avenue", "r => rue", "rte => route" ]
        }
      }
    }
  }, // ...
```

Index template (mapping)

```
{
  "template": ".bano-*", "settings": { ... },
  "mappings": {
    "doc": {
      "properties" : {
        "address": {
          "properties" : {
            "city": {
              "type": "text", "analyzer": "bano_analyzer",
              "fields": { "keyword": { "type": "keyword" } }
            },
            "number": { "type": "keyword" },
            "street_name": { "type": "text", "analyzer": "bano_street_analyzer" },
            "zipcode": { "type": "keyword" }
          }
        },
        "region": { "type": "keyword" },
        "location": { "type": "geo_point" },
        "id": { "type": "keyword" },
        "source": { "type": "keyword" }
}}}, // ...
```

Index template (aliases)

```
{
  "template": ".bano-*",
  "settings": { ... },
  "mappings": { ... },
  "aliases" : {
    ".bano" : {}
  }
}
```

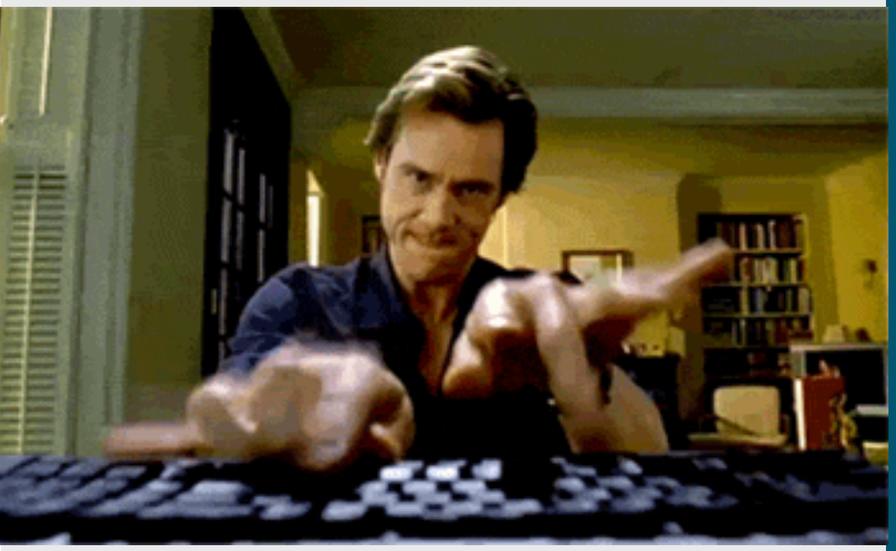


Launch Logstash

```
export SOURCE_DIR=~/Documents/ingest-bano/
DATASOURCE_DIR=~/Documents/ingest/bano-data
LOGSTASH=~/Documents/ingest/stack-6.0.0/logstash-6.0.0

import_region () {
    export REGION=$1
    FILE=$DATASOURCE_DIR/bano-$REGION.csv
    curl -XDELETE localhost:9200/.bano-$REGION?pretty
    cat $FILE | $LOGSTASH/bin/logstash -f $SOURCE_DIR/src/main/logstash/import.conf
}

DEPTS=95
for i in {01..19} $(seq 21 $DEPTS) {971..974} {976..976} ; do
    DEPT=$(printf %02d $i)
    import_region $DEPT
done
```



Writing an ingest plugin

elastic

Use bano processor

elastic

elastic

Ingest Node

(re)indexing and enriching documents within Elasticsearch

David Pilato
Developer | Evangelist, @dadoonet

Watch this space: https://github.com/dadoonet
And follow me on Twitter!