



Ingest Node

(re)indexing and enriching documents within Elasticsearch



David Pilato
Developer | Evangelist
@dadoonet



is powered by

elastic
User Group

Feb 7, 2010 – Sep 5, 2016

Contributions to master, excluding merge commits

Contributions: Commits ▾



kimchy

#1

4,737 commits / 1,401,534 ++ / 1,064,899 --



s1monw

#2

2,216 commits / 1,219,296 ++ / 1,217,559 --



dadoonet

#3

1,171 commits / 128,081 ++ / 84,741 --



javanna

#4

1,101 commits / 142,713 ++ / 119,744 --



martijnvg

#5

1,093 commits / 212,940 ++ / 142,674 --



clintongormley

#6

854 commits / 232,442 ++ / 30,976 --



rmuir

#7

833 commits / 161,166 ++ / 147,023 --



jasontedor

#8

772 commits / 62,452 ++ / 37,875 --



jpountz

#9

737 commits / 173,140 ++ / 204,220 --

Elastic Stack

User Interface



Store, Index,
& Analyze



Ingest



X-Pack

Security

Alerting

Monitoring

Reporting

Graph

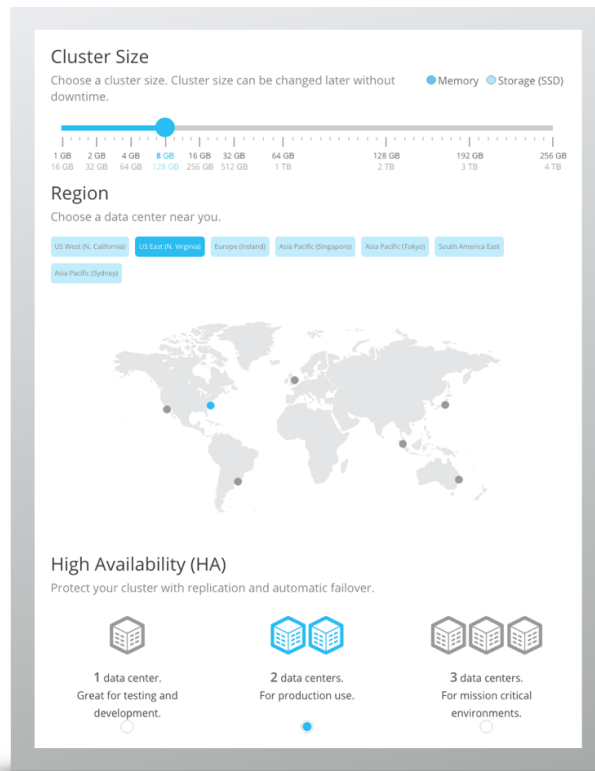


Elastic Cloud



The only Elasticsearch as a Service offering powered by the creators of the Elastic Stack

- Always runs on the latest software
- One-click to scale/upgrade with no downtime
- Free Kibana and backups every 30 minutes
- Dedicated, SLA-based support
- Easily add X-Pack features: security (Shield), alerting (Watcher), and monitoring (Marvel)
- Pricing starts at \$45 a month



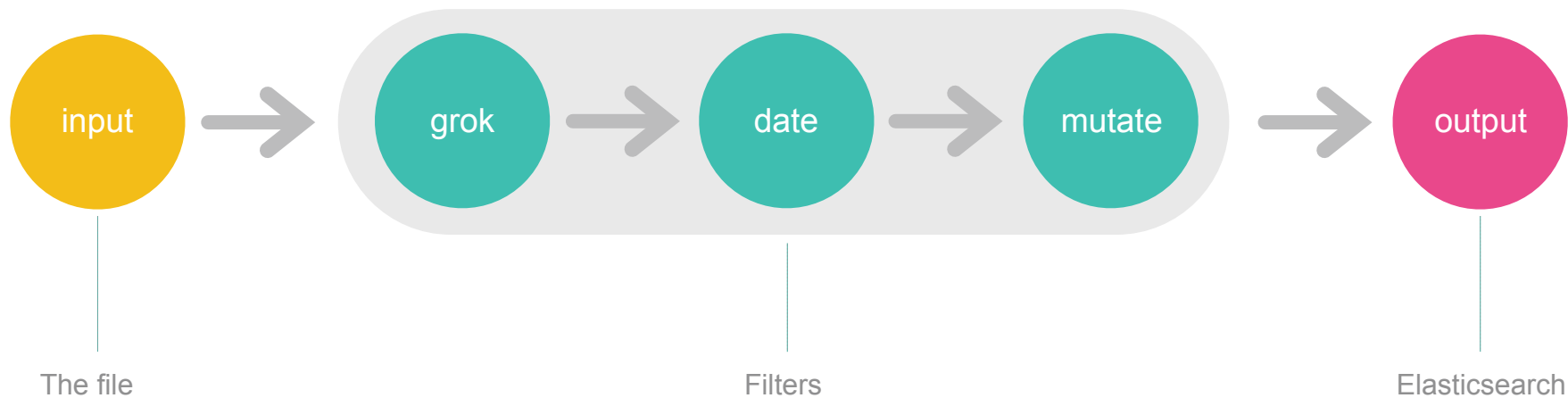


Why ingest node?

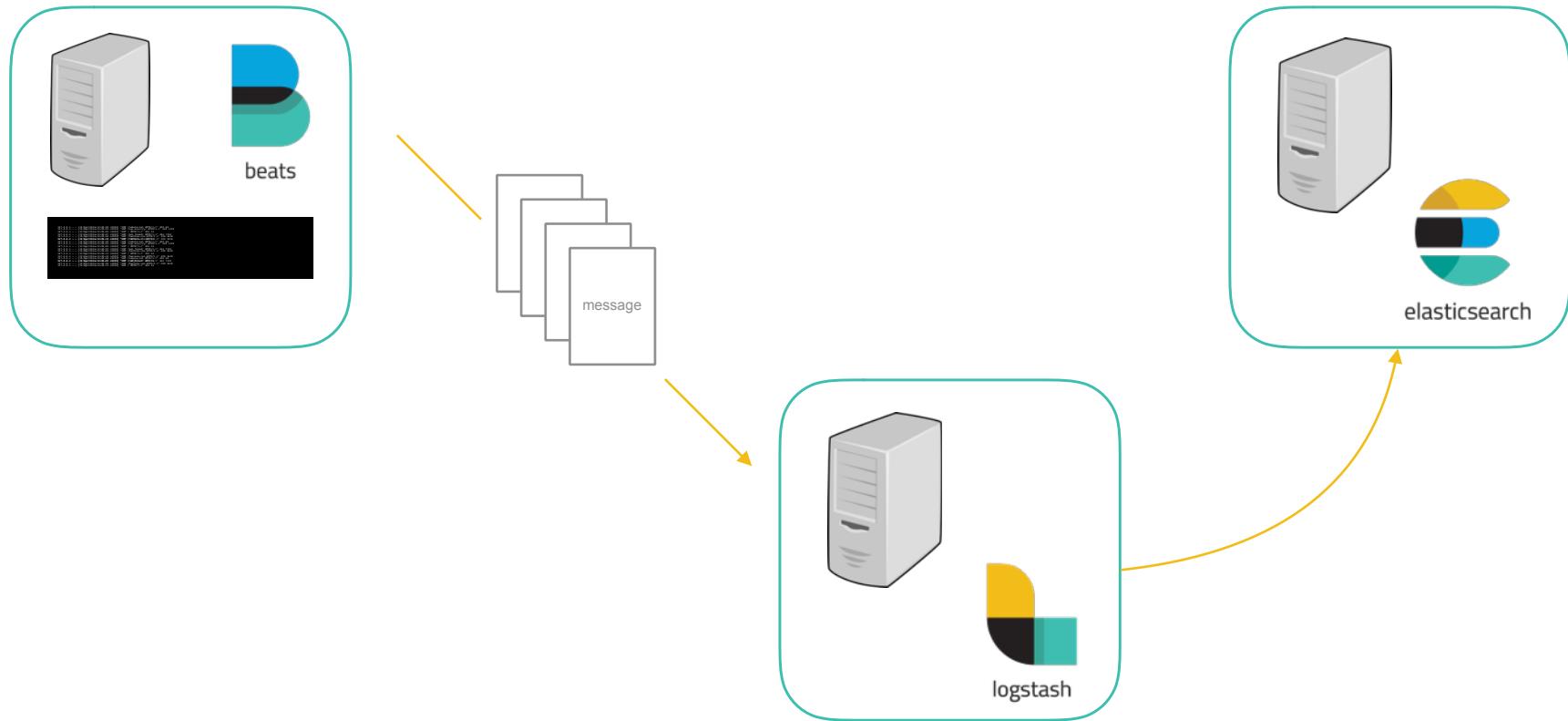


I just want to tail a file.

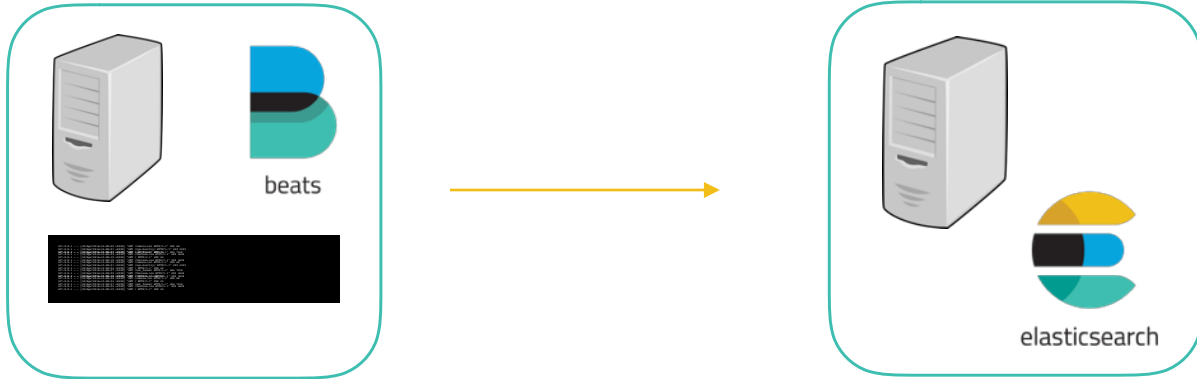
Logstash: collect, enrich & transport



Logstash common setup



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 +0200] "GET /favicon.ico HTTP/1.1" 200 3638
```



beats

```
{
  "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 +0200] \"GET /favicon.ico HTTP/1.1\" 200 3638"
}
```



elasticsearch

Elasticsearch: enrich and index

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

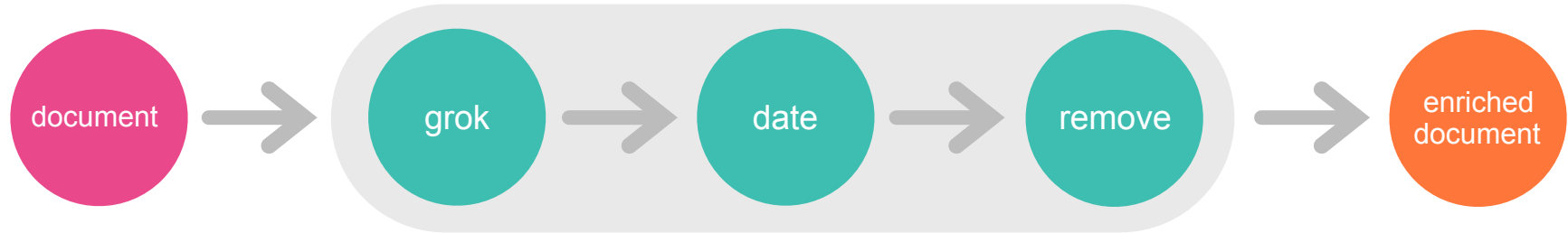


```
{  
  "request" : "/",  
  "auth" : "-",  
  "ident" : "-",  
  "verb" : "GET",  
  "@timestamp" : "2016-04-19T10:00:04.000Z",  
  "response" : "200",  
  "bytes" : "24",  
  "clientip" : "127.0.0.1",  
  "httpversion" : "1.1",  
  "rawrequest" : null,  
  "timestamp" : "19/Apr/2016:12:00:04 +0200"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



How does ingest node work?

Ingest pipeline



Pipeline: a set of processors

Define a pipeline

```
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"
      }
    }
  ]
}
```

Index a document

Provide the id of the pipeline to execute

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

What has actually been indexed

```
GET /logs/apache/1
```

```
{  
  "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"  
}
```

Pipeline management

Create, Read, Update & Delete

```
PUT /_ingest/pipeline/apache-log
{
  ...
}
```

```
GET /_ingest/pipeline/apache-log
```

```
GET /_ingest/pipeline/*
```

```
DELETE /_ingest/pipeline/apache-log
```

grok uppercase split 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"  
  }  
}
```

Foreach processor

Do something for every element of an array

```
{
  "foreach": {
    "field" : "values",
    "processor" : {
      "uppercase" : {
        "field" : "_ingest._value"
      }
    }
  }
}
```

Fail processor

Raises an exception with a configurable message

```
{  
  "fail": {  
    "message": "custom error"  
  }  
}
```

Plugins

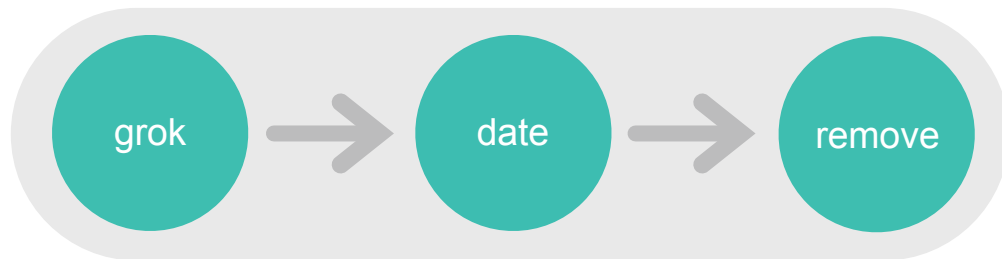
Introducing new processors is as easy as writing a plugin

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

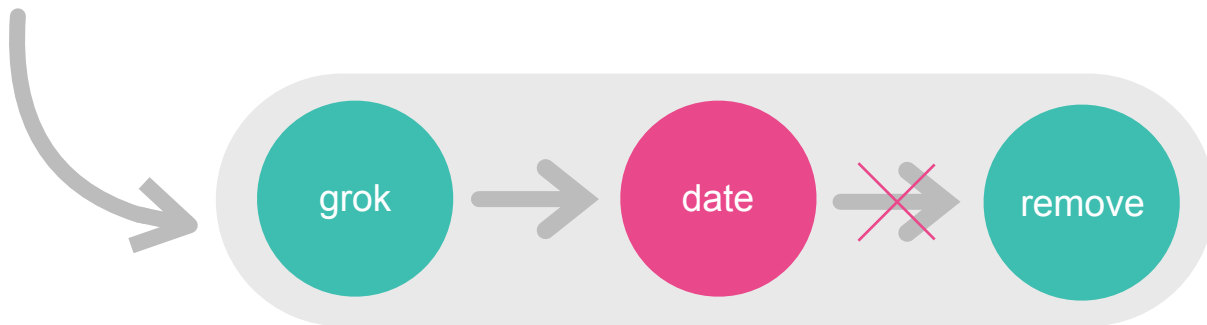


Error handling

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

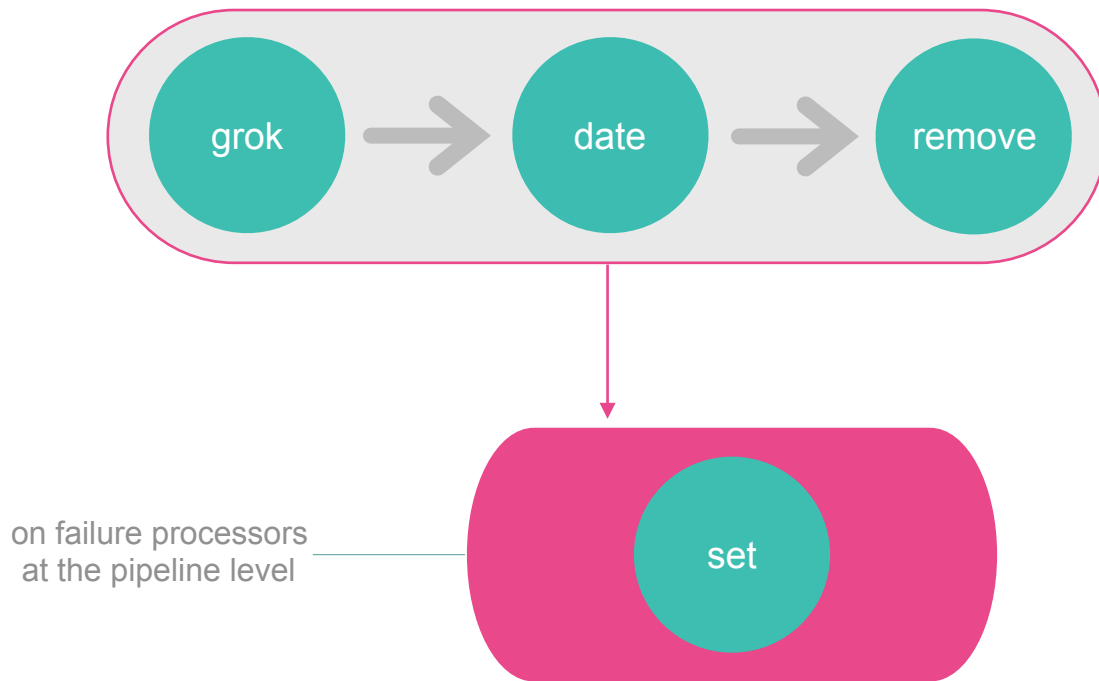


```
{  
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \"GET / HTTP/1.1\" 200 24"  
}
```

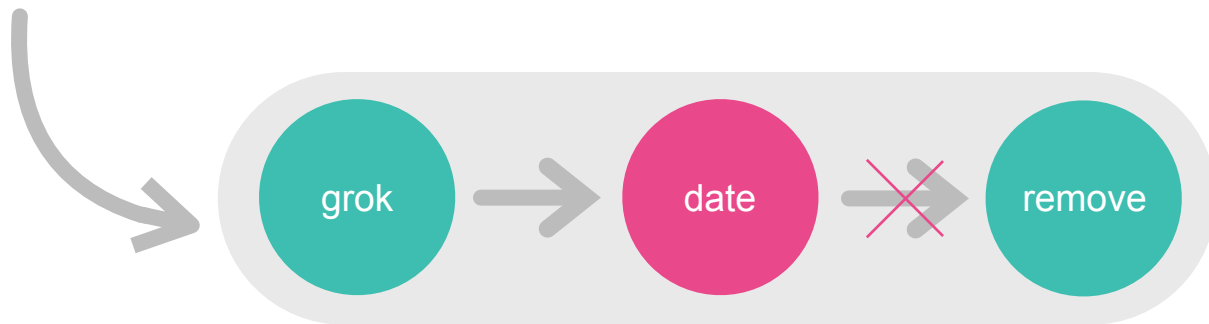


400 Bad Request
unable to parse date [19/Apr/2016:12:00:00 +040]

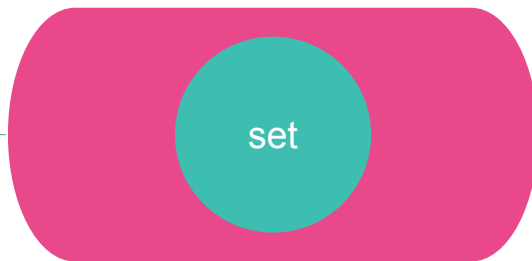
```
{  
  "message" : "127.0.0.1 - - [19/Apr/2016:12:00:00 +040] \\"GET / HTTP/1.1\\" 200 24"  
}
```



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

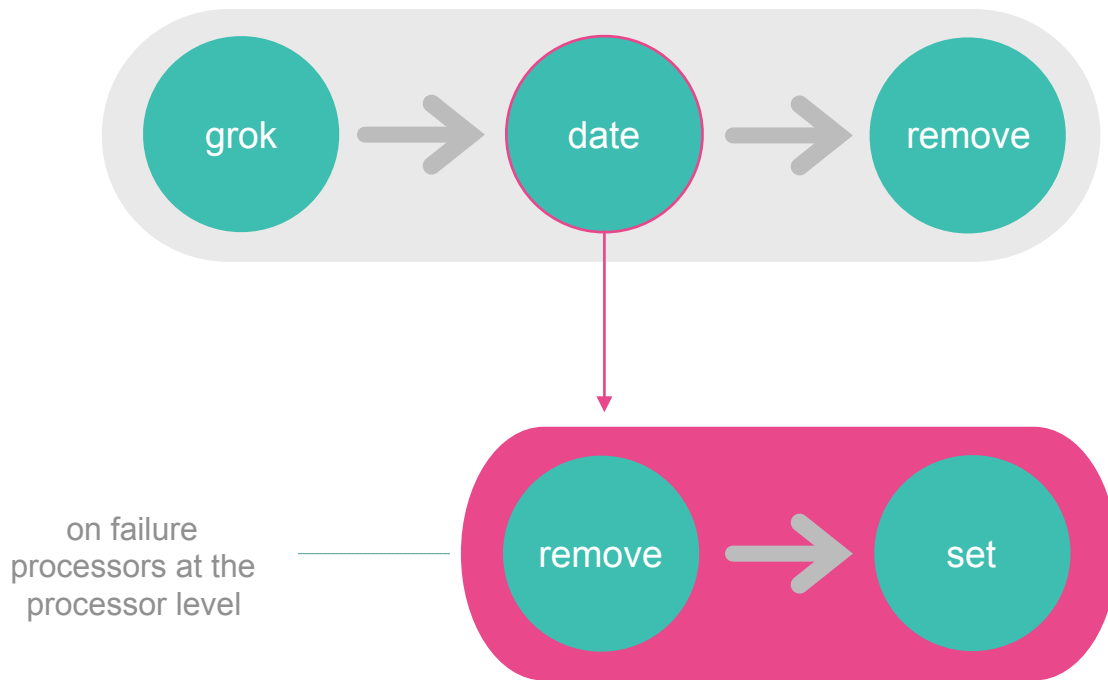


on failure processors
at the pipeline level

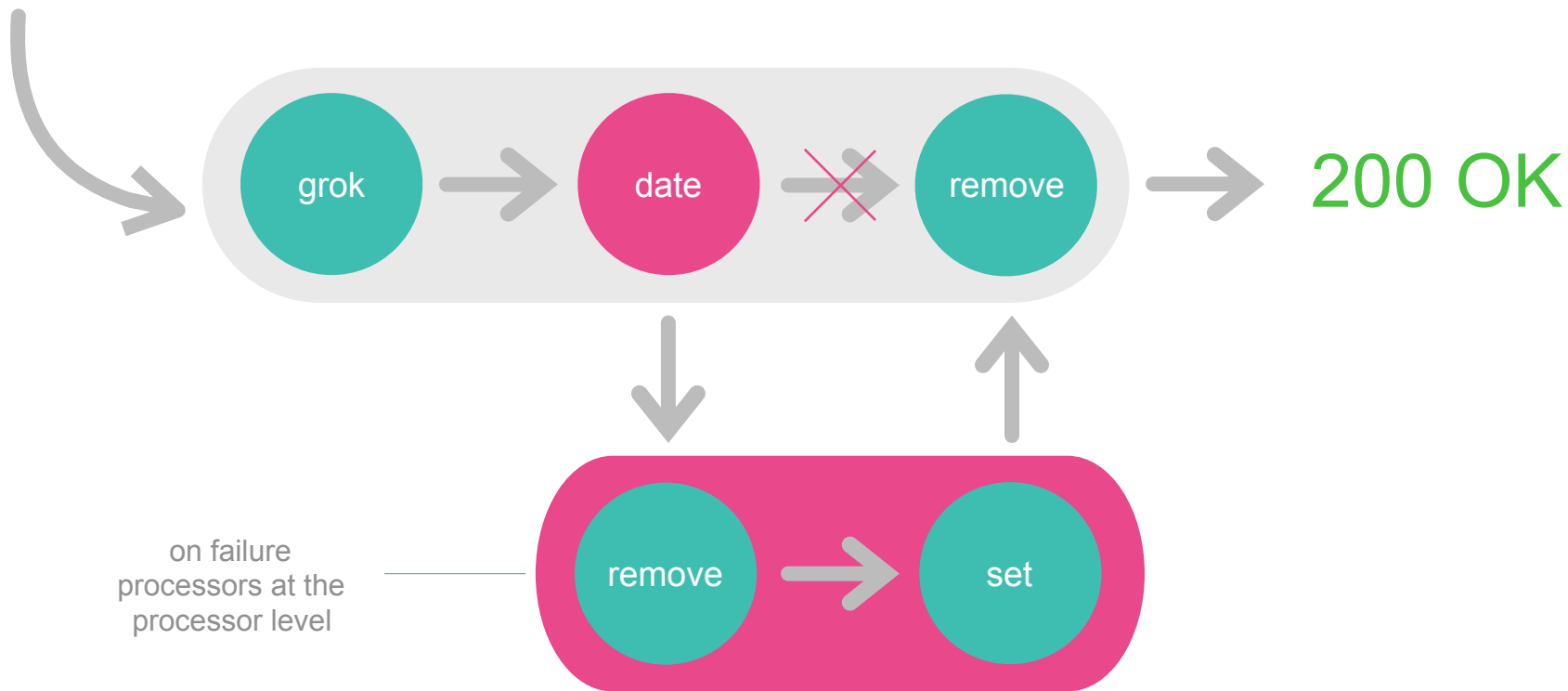


200 OK

```
{  
  "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 +0400] \"GET / HTTP/1.1\" 200 24"  
}
```



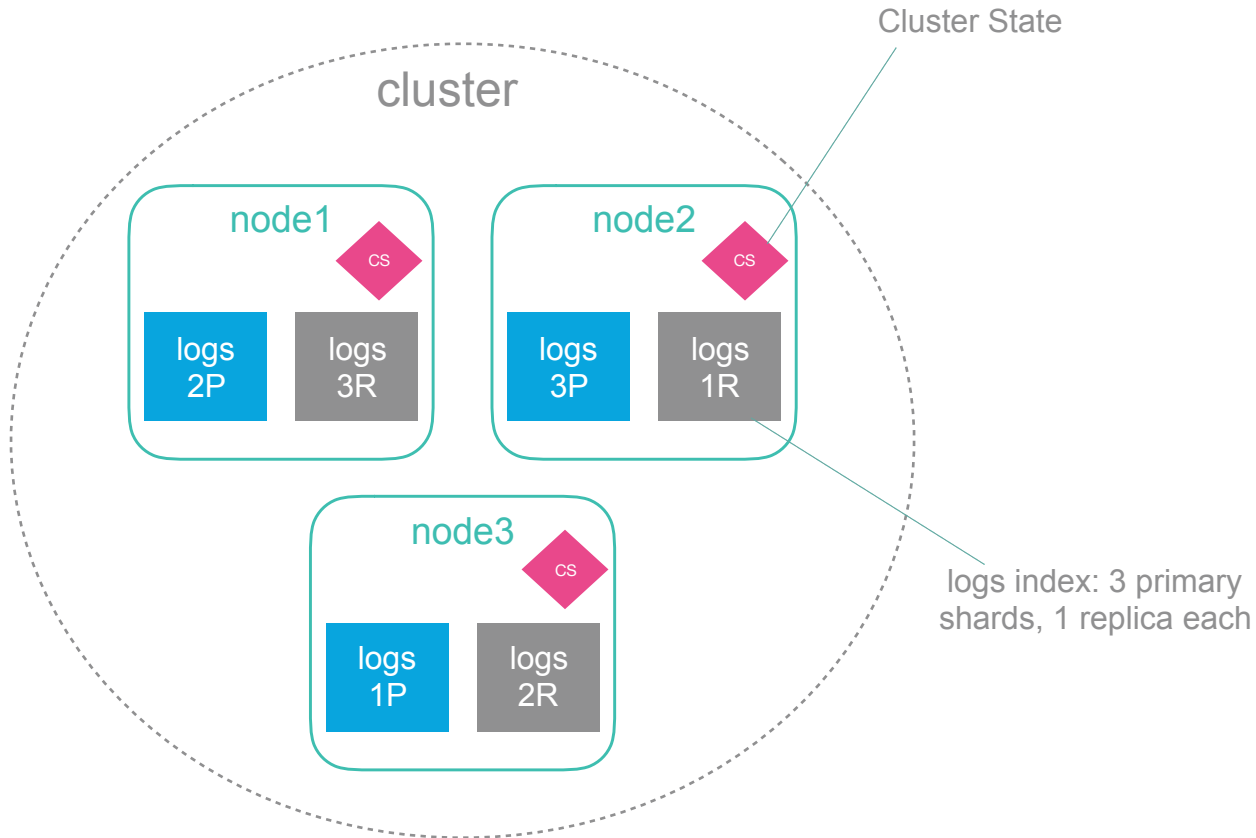


Ingest node internals

Default scenario

All nodes are equal:

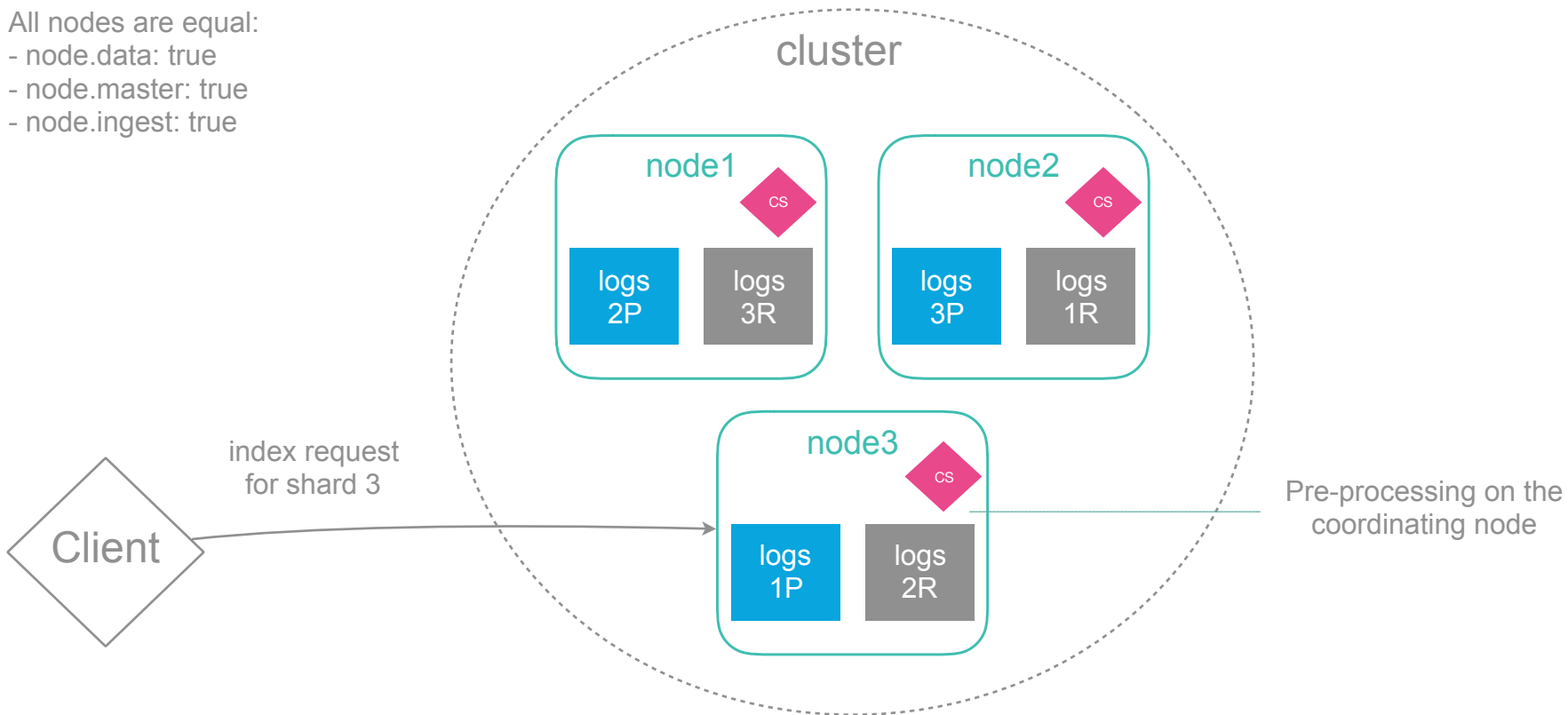
- node.data: true
- node.master: true
- node.ingest: true



Default scenario

All nodes are equal:

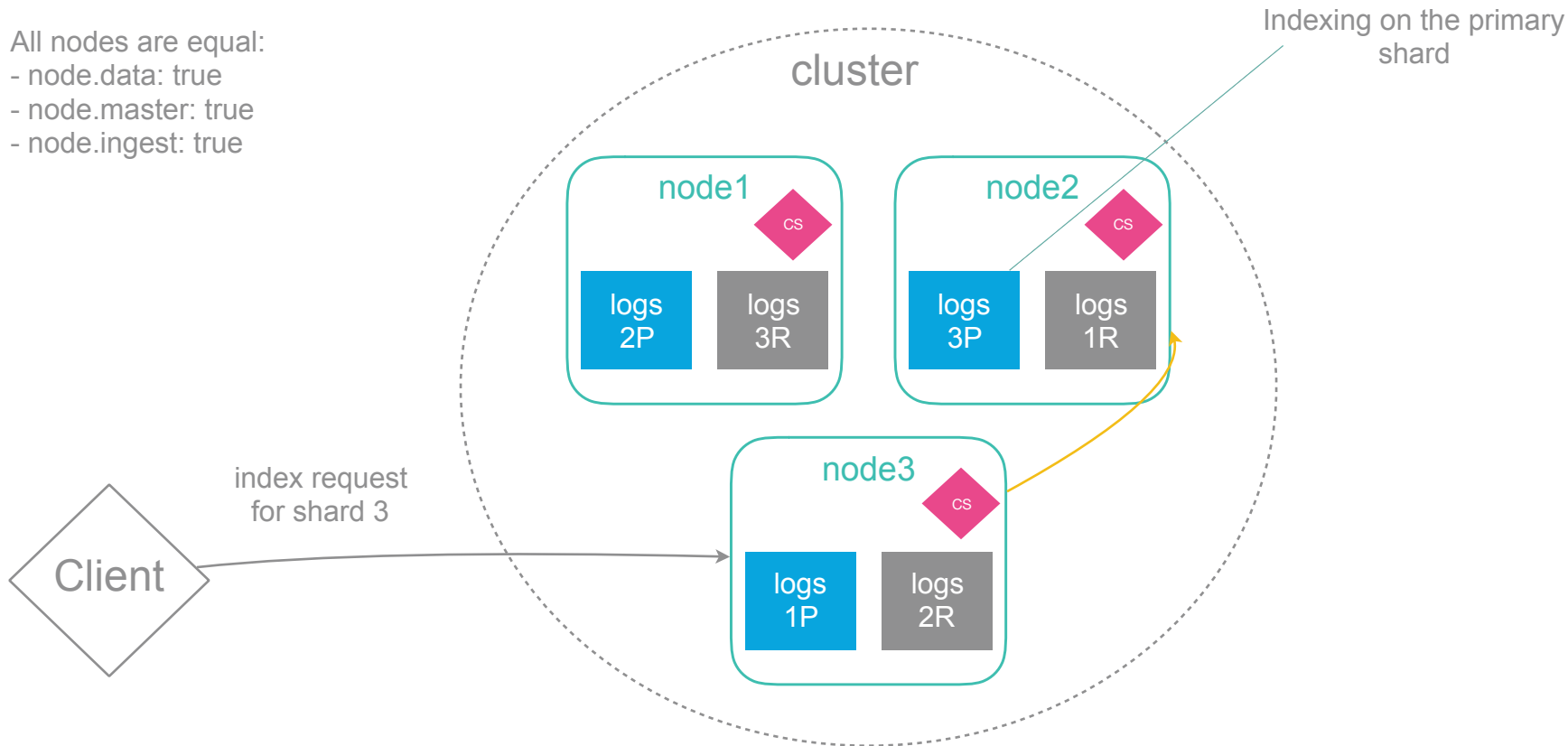
- node.data: true
- node.master: true
- node.ingest: true



Default scenario

All nodes are equal:

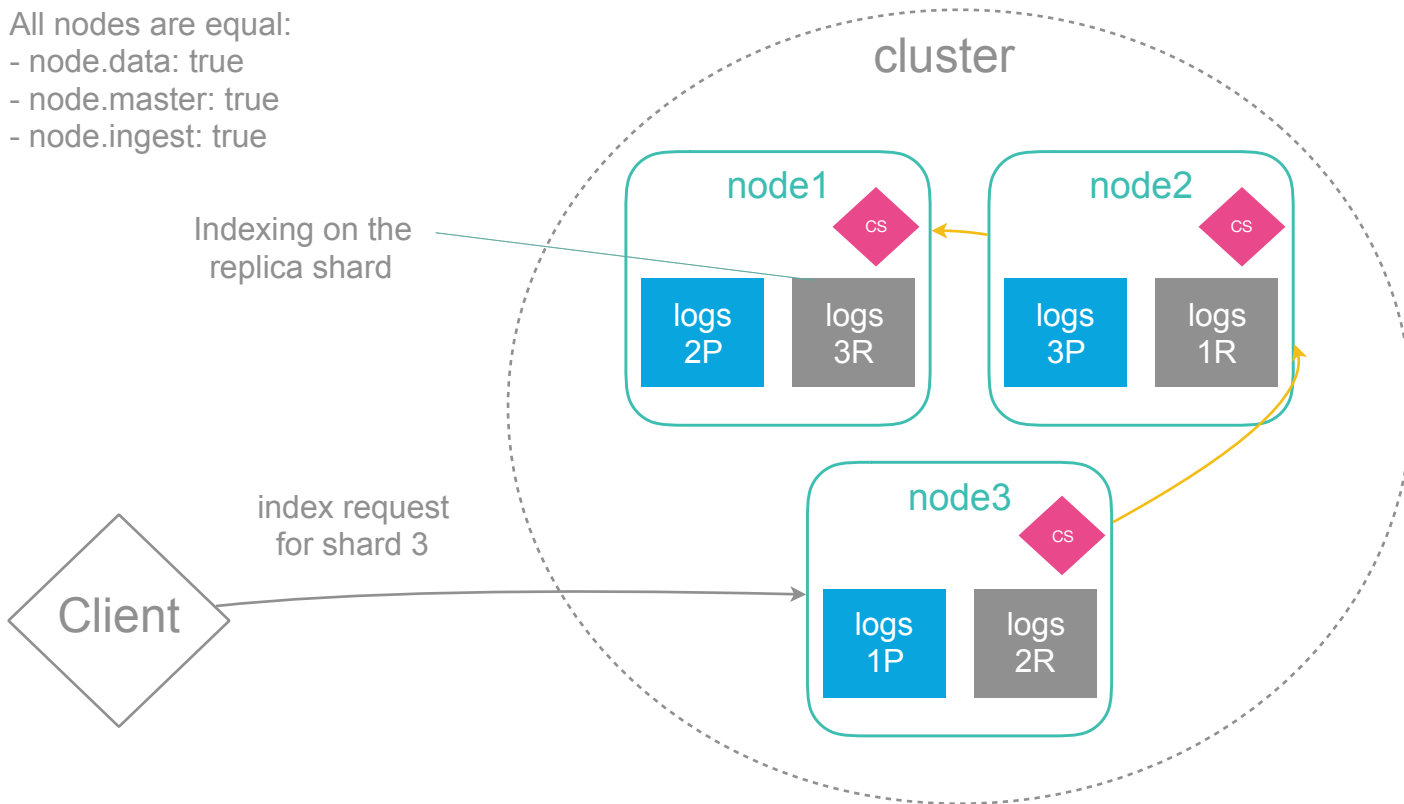
- node.data: true
- node.master: true
- node.ingest: true



Default scenario

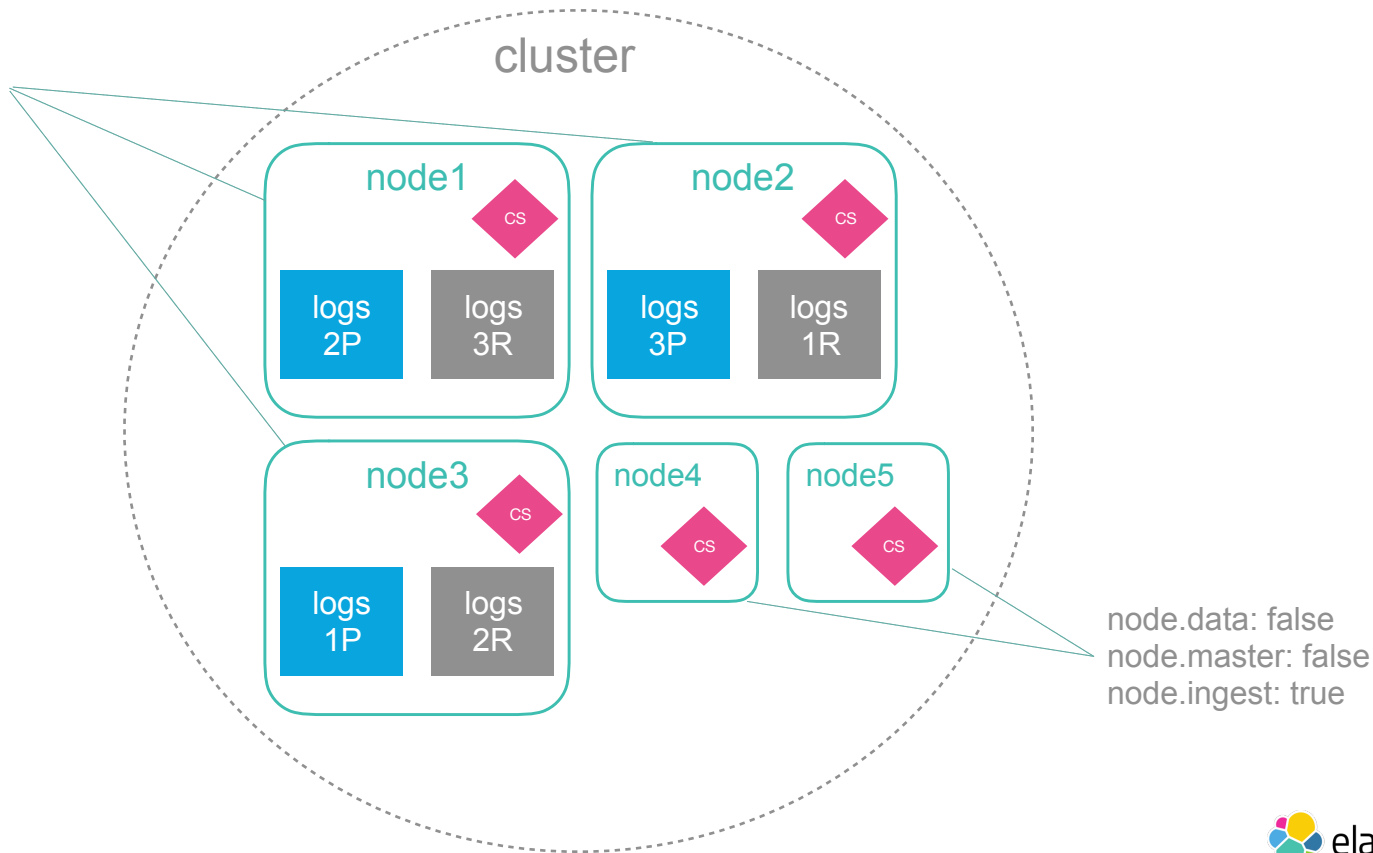
All nodes are equal:

- node.data: true
- node.master: true
- node.ingest: true

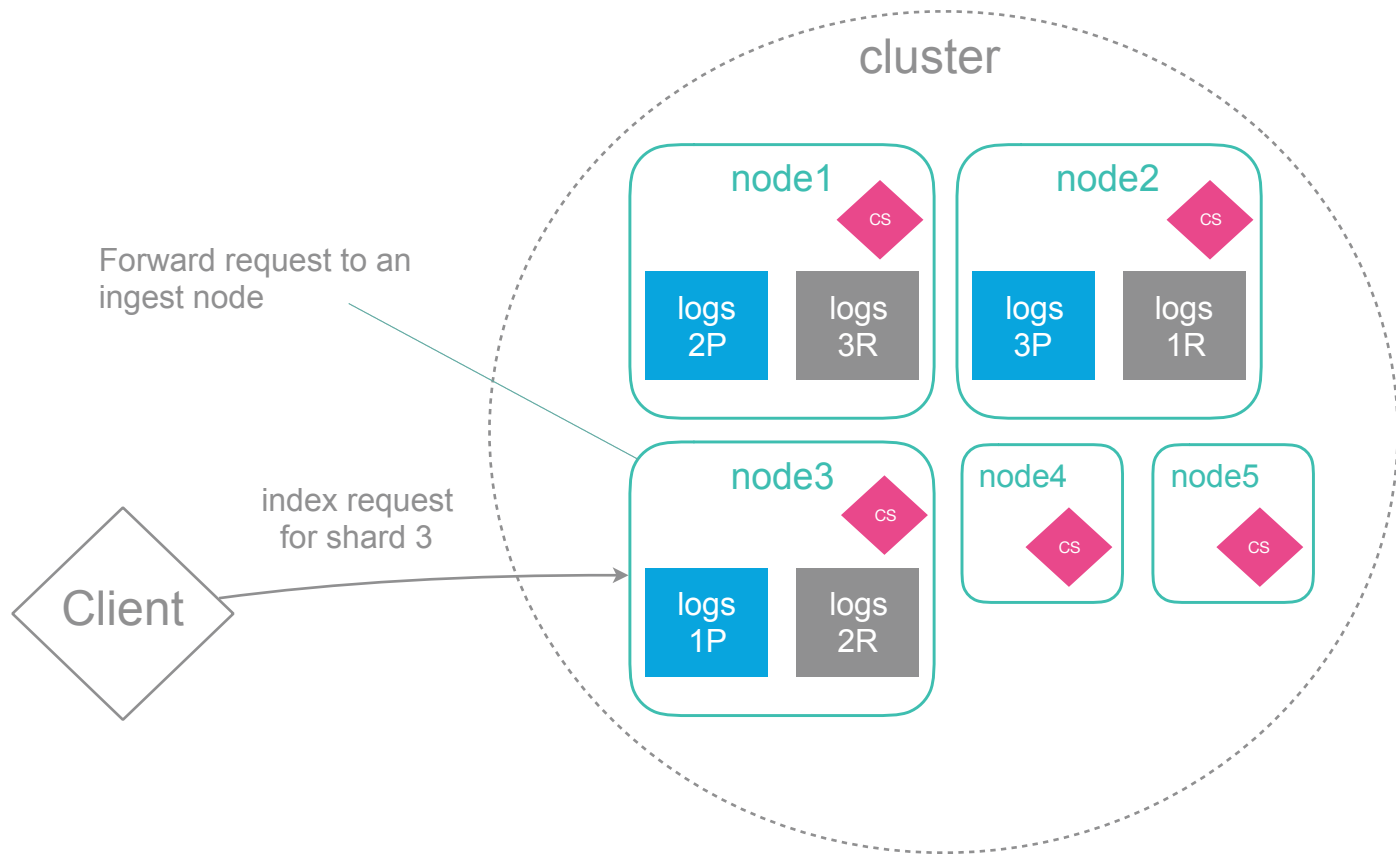


Ingest dedicated nodes

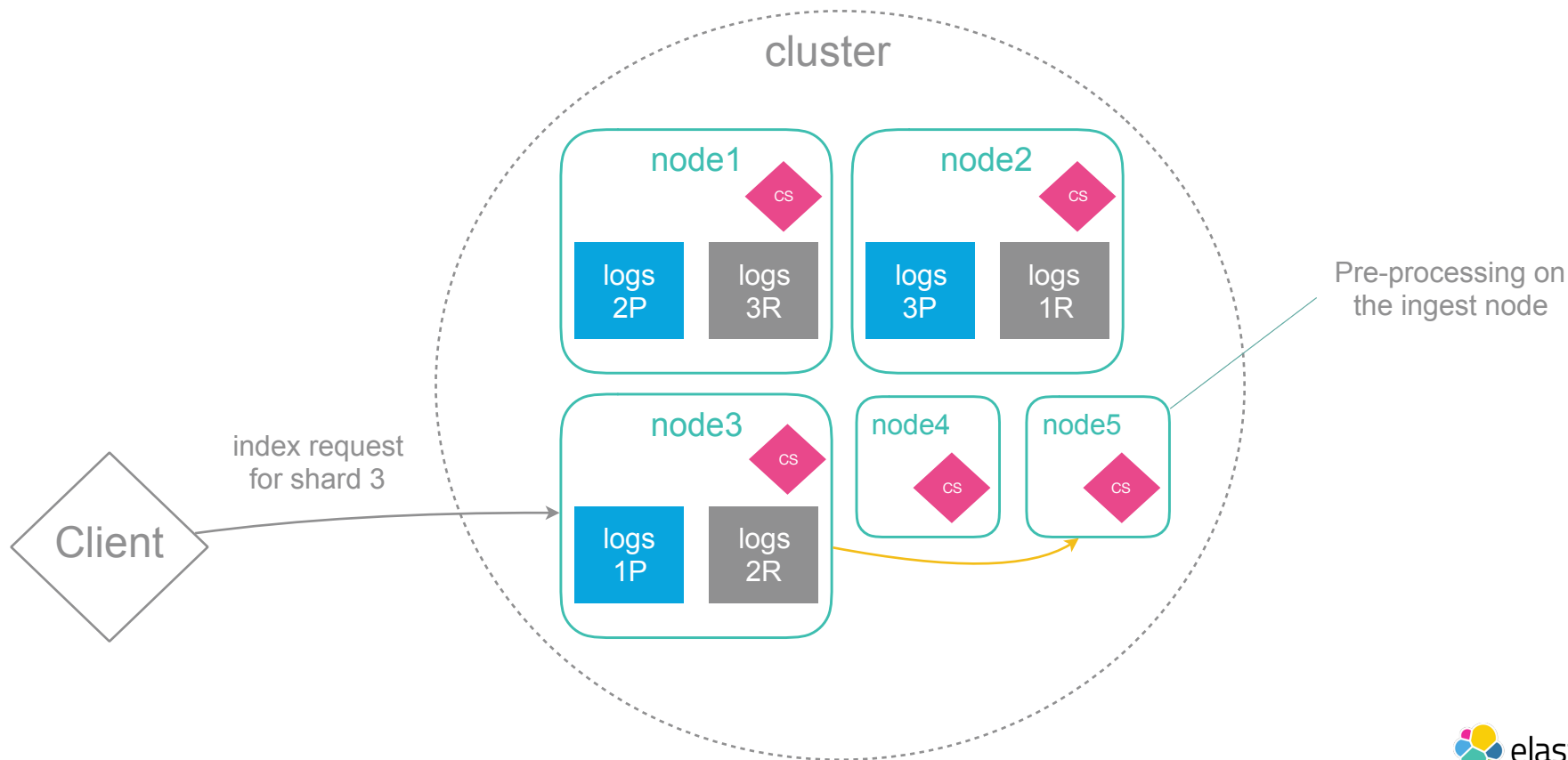
node.data: true
node.master: true
node.ingest: false



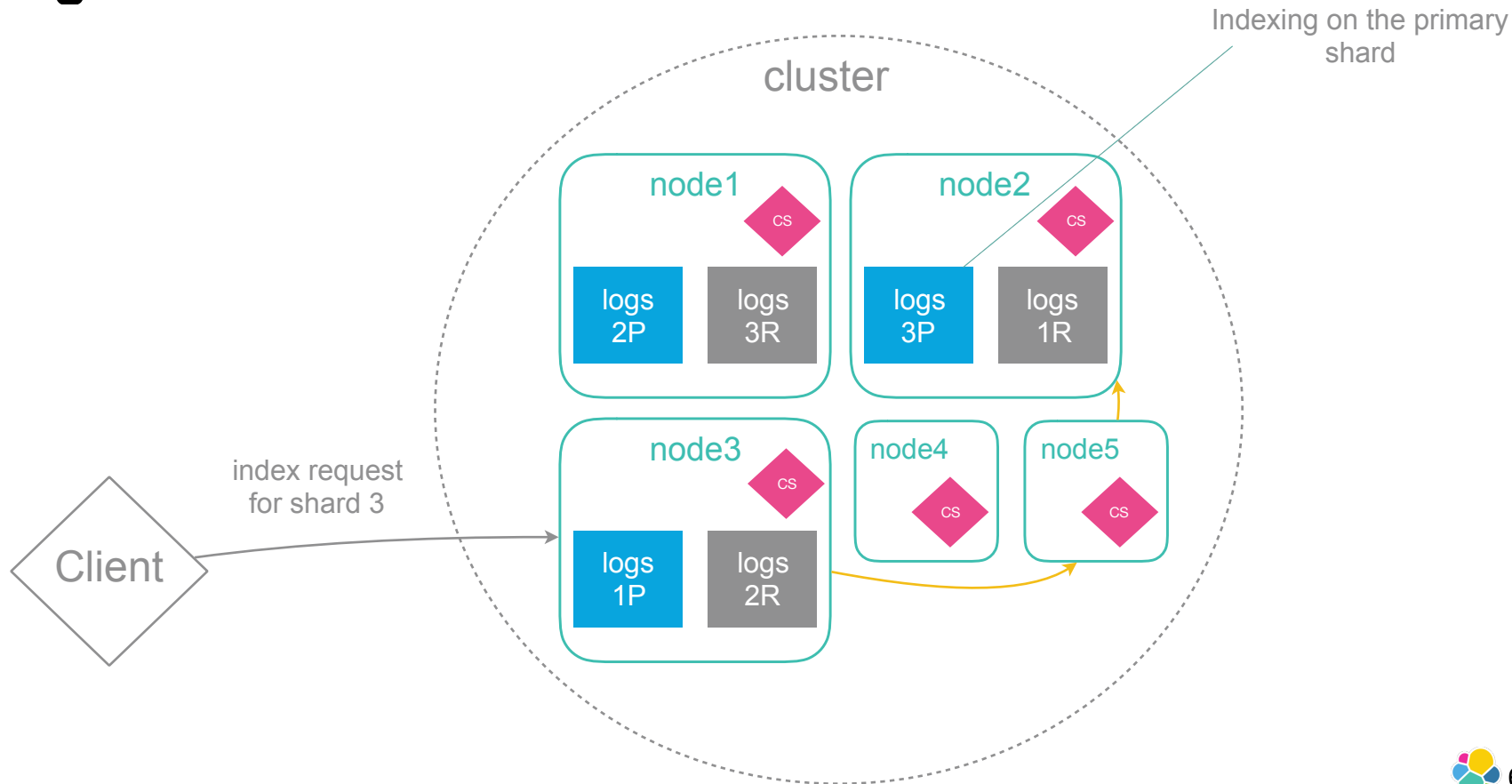
Ingest dedicated nodes



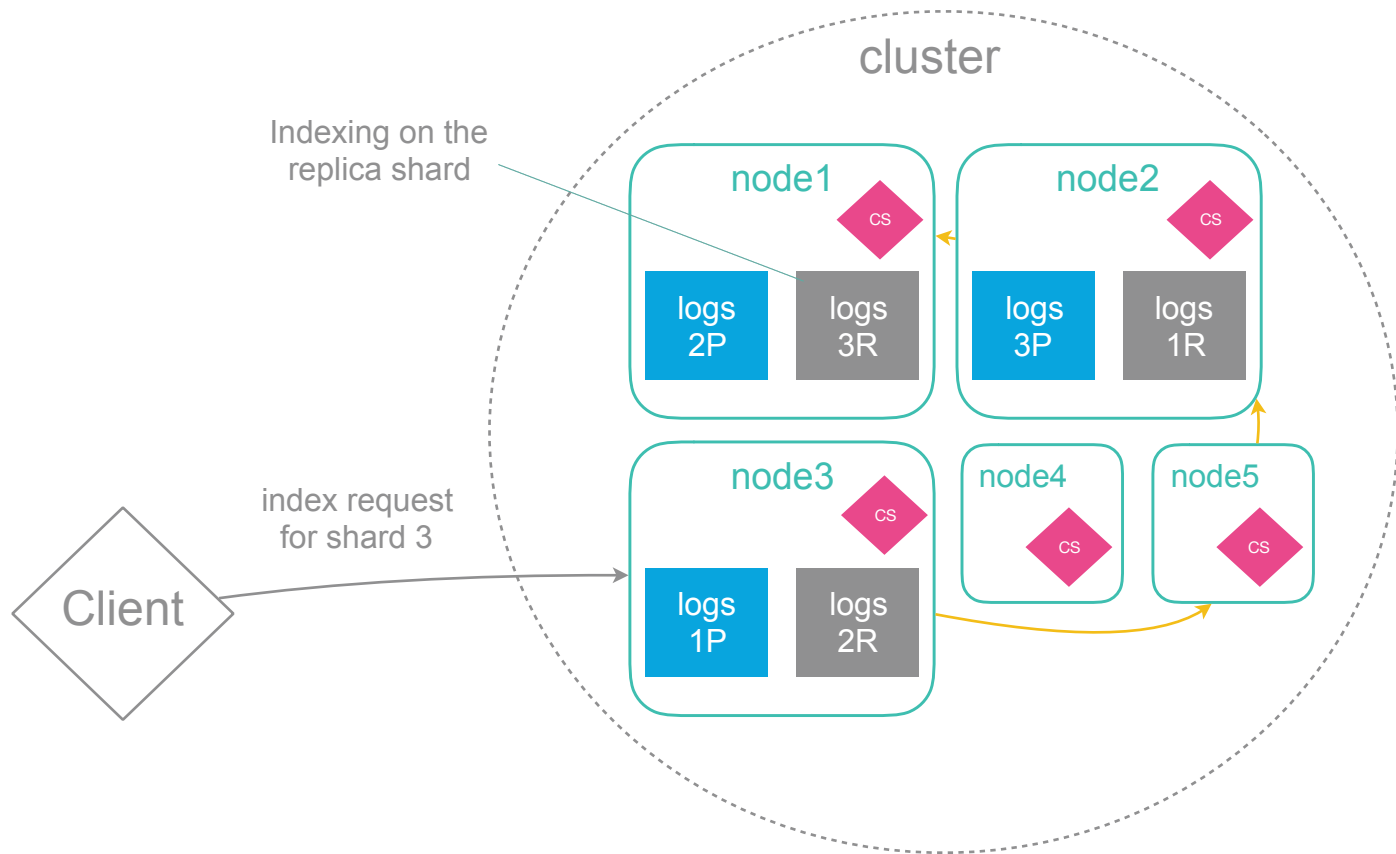
Ingest dedicated nodes



Ingest dedicated nodes



Ingest dedicated nodes





Where can ingest pipelines be used?

Index api

```
PUT /logs/apache/1?pipeline=apache-log
{
  "message" : "..."
```

Bulk api

```
PUT /logs/_bulk
{ "index": { "_type": "apache", "_id":
"1", "pipeline": "apache-log" } } \n
{ "message" : "..." } \n
{ "index": { "_type": "mysql", "_id":
"1", "pipeline": "mysql-log" } } \n
{ "message" : "..." } \n
```

Reindex api

Scroll & bulk indexing
made easy

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



bano-ingest plugin

From postal address to geo_point
From geo_point to postal address

What is BANO?

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

per department

all addresses

 bano-95-shp.zip	23-May-2016 06:48	8.6M
 bano-95.csv	23-May-2016 06:47	16M
 bano-95.json.gz	23-May-2016 07:31	2.3M
 bano-95.ttl.gz	23-May-2016 06:47	7.1M
 bano-971-shp.zip	23-May-2016 06:46	1.9M
 bano-971.csv	23-May-2016 06:46	3.7M
 bano-971.json.gz	23-May-2016 07:29	525K
 bano-971.ttl.gz	23-May-2016 06:46	1.6M
 bano-972-shp.zip	23-May-2016 06:46	1.6M
 bano-972.csv	23-May-2016 06:46	3.3M
 bano-972.json.gz	23-May-2016 07:30	473K
 bano-972.ttl.gz	23-May-2016 06:46	1.4M
 bano-973-shp.zip	23-May-2016 06:46	630K
 bano-973.csv	23-May-2016 06:46	1.1M
 bano-973.json.gz	23-May-2016 07:29	189K
 bano-973.ttl.gz	23-May-2016 06:46	521K
 bano-974-shp.zip	23-May-2016 06:49	8.7M
 bano-974.csv	23-May-2016 06:48	17M
 bano-974.json.gz	23-May-2016 07:31	2.2M
 bano-974.ttl.gz	23-May-2016 06:49	7.3M
 bano-976-shp.zip	23-May-2016 06:46	241K
 bano-976.csv	23-May-2016 06:46	441K
 bano-976.json.gz	23-May-2016 07:29	78K
 bano-976.ttl.gz	23-May-2016 06:46	202K
 code_cadastre.csv	23-May-2016 00:05	1.3M
 full.csv.gz	23-May-2016 06:50	198M
 full.sjson.gz	23-May-2016 07:34	249M

BANO Format

- bano-976.csv sample (full.csv.gz has same format)

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

Features

- Download, transform and index BANO datasource

```
curl -XPUT 127.0.0.1:9200/_bano/17
curl -XPUT 127.0.0.1:9200/_bano/17,95,29
curl -XPUT 127.0.0.1:9200/_bano/_full
```

- Create a new ingest processor

```
curl -XPUT "localhost:9200/_ingest/pipeline/bano-test?pretty" -d '{
  "description": "my_pipeline",
  "processors": [ {
    "bano": {}
  } ]
}'
```

From structured address (french format)...

```
curl -XPOST "localhost:9200/_ingest/pipeline/bano-test/_simulate?pretty" -d '{
  "docs": [ {
    "_index": "index",
    "_type": "type",
    "_id": "id",
    "_source": {
      "address": {
        "number": "25",
        "street_name": "georges",
        "zipcode": "17440",
        "city": "Aytré"
      }
    }
  } ]
}'
```

To normalized address with coordinates...

```
"doc" : {  
  "_source" : {  
    "address" : {  
      "zipcode" : "17440",  
      "number" : "25",  
      "city" : "Aytré",  
      "street_name" : "georges"  
    },  
    "bano_address" : {  
      "zipcode" : "17440",  
      "number" : "25",  
      "city" : "Aytré",  
      "street_name" : "Boulevard Georges Clemenceau",  
      "full_address" : "25, Boulevard Georges Clemenceau 17440 Aytré",  
      "location" : {  
        "lon" : -1.122966,  
        "lat" : 46.130368  
      }  
    }  
  }  
}
```

From a geo point...

```
curl -XPOST "localhost:9200/_ingest/pipeline/bano-test/_simulate?pretty" -d '{
  "docs": [ {
    "_index": "index",
    "_type": "type",
    "_id": "id",
    "_source": {
      "location": {
        "lat": 46.135283,
        "lon": -1.113750
      }
    }
  } ]
}'
```

To the closest full address...

```
"doc" : {  
  "_source" : {  
    "location" : {  
      "lon" : -1.11375,  
      "lat" : 46.135283  
    },  
    "bano_address" : {  
      "zipcode" : "17440",  
      "number" : "1",  
      "city" : "Aytré",  
      "street_name" : "Rue du Petit Versailles",  
      "full_address" : "1, Rue du Petit Versailles 17440 Aytré",  
      "location" : {  
        "lon" : -1.113564,  
        "lat" : 46.135343  
      }  
    }  
  }  
}
```

Combine with other ingest processors

```
curl -XPUT "localhost:9200/_ingest/pipeline/bano-test-4?pretty" -d '{
  "description": "debug",
  "processors": [ {
    "geoip" : {
      "field" : "ip"
    }
  }, {
    "bano": {
      "location_lat_field": "geoip.location.lat",
      "location_lon_field": "geoip.location.lon"
    }
  } ]
}'
```

From an IP address...

```
curl -XPOST "localhost:9200/_ingest/pipeline/bano-test-4/_simulate?pretty&verbose" -d '{
  "docs": [ {
    "_index": "index",
    "_type": "type",
    "_id": "id",
    "_source": {
      "ip" : "82.229.80.187"
    }
  } ]
}'
```

To the closest full address...

```
"doc" : {  
  "_source" : {  
    "ip" : "82.229.80.187",  
    "geoip" : {  
      "continent_name" : "Europe", "city_name" : "Cergy",  
      "country_iso_code" : "FR", "region_name" : "Val d'Oise",  
      "location" : { "lon" : 2.0761, "lat" : 49.0364 }  
    },  
    "bano_address" : {  
      "zipcode" : "95000",  
      "number" : "3",  
      "city" : "Cergy",  
      "location" : {  
        "lon" : 2.075687,  
        "lat" : 49.037202  
      },  
      "full_address" : "3, Avenue des Trois Fontaines 95000 Cergy",  
      "street_name" : "Avenue des Trois Fontaines"  
    }  
  }  
}
```

Writing the processor

```
public final class BanoProcessor extends AbstractProcessor {
    private final String cityField;
    public BanoProcessor(String cityField) { this.cityField = cityField; }

    @Override
    public void execute(IngestDocument ingestDocument) {

        // Implement your logic code here
        if (ingestDocument.hasField(cityField)) {
            String city = ingestDocument.getFieldValue(cityField, String.class)

            // Like searching in elasticsearch with a city field
            Location location = banoEsClient.searchByCity(city);

            // Then modify the document as you wish
            Map<String, Object> locationObject = new HashMap<>();
            locationObject.put("lat", location.getLat());
            locationObject.put("lon", location.getLon());
            ingestDocument.setFieldValue("location", locationObject);
        }
    }
}
```

Writing the processor factory

```
public static final class Factory implements Processor.Factory {
    @Override
    public Processor create(Map<String, Processor.Factory> map, String processorTag,
                           Map<String, Object> config) throws Exception {
        // Read the bano processor config
        String cityField = readStringProperty("bano", processorTag, config,
            "city_field", // We read here the value of "city_field" in config
            "address.city"); // If not set we will read from "address.city" by default

        // Do the same for other fields

        // Create the processor instance
        return new BanoProcessor(cityField, otherFields...);
    }
}
```

Writing an ingest plugin

```
public class IngestBanoPlugin extends Plugin implements IngestPlugin {  
    @Override  
    public Map<String, Processor.Factory> getProcessors(Processor.Parameters parameters) {  
        return Collections.singletonMap("bano", new BanoProcessor.Factory());  
    }  
}
```



Demo time!



Go get Elasticsearch 5.0.0!

<https://www.elastic.co/downloads/elasticsearch>





David Pilato

Developer | Evangelist

@dadoonet

Bano ingest plugin

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

