

Introducción a ELK

Adrián Santos Marrero <adsaman@gmail.com>



elasticsearch.



logstash



Flujo de datos



Data



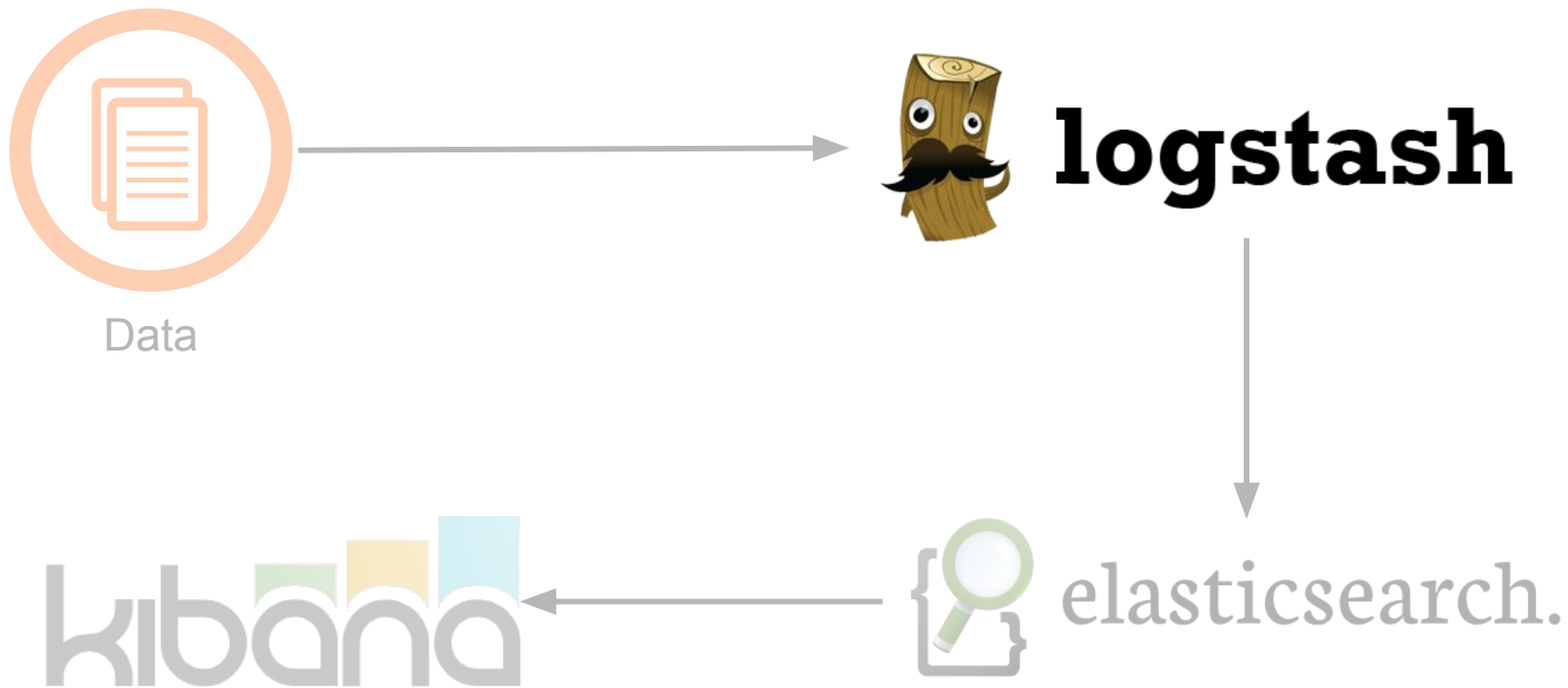
logstash



elasticsearch.



Flujo de datos



Logstash

- Gestiona eventos y logs
- Recolecta
- Analiza
- Enriquece
- Almacena datos
- Software libre: Licencia Apache 2.0



Arquitectura de logstash

Input

datastore
stream
log files
files
monitoring
queue
network



Filter



parse, enrich, tag, drop



Output

datastore
files
e-mail
pager
monitoring
chat
API
queues

Arquitectura de logstash

Input

datastore
stream
log files
files
monitoring
queue
network

ip: 193.145.120.40



Filter



parse, enrich, tag, drop

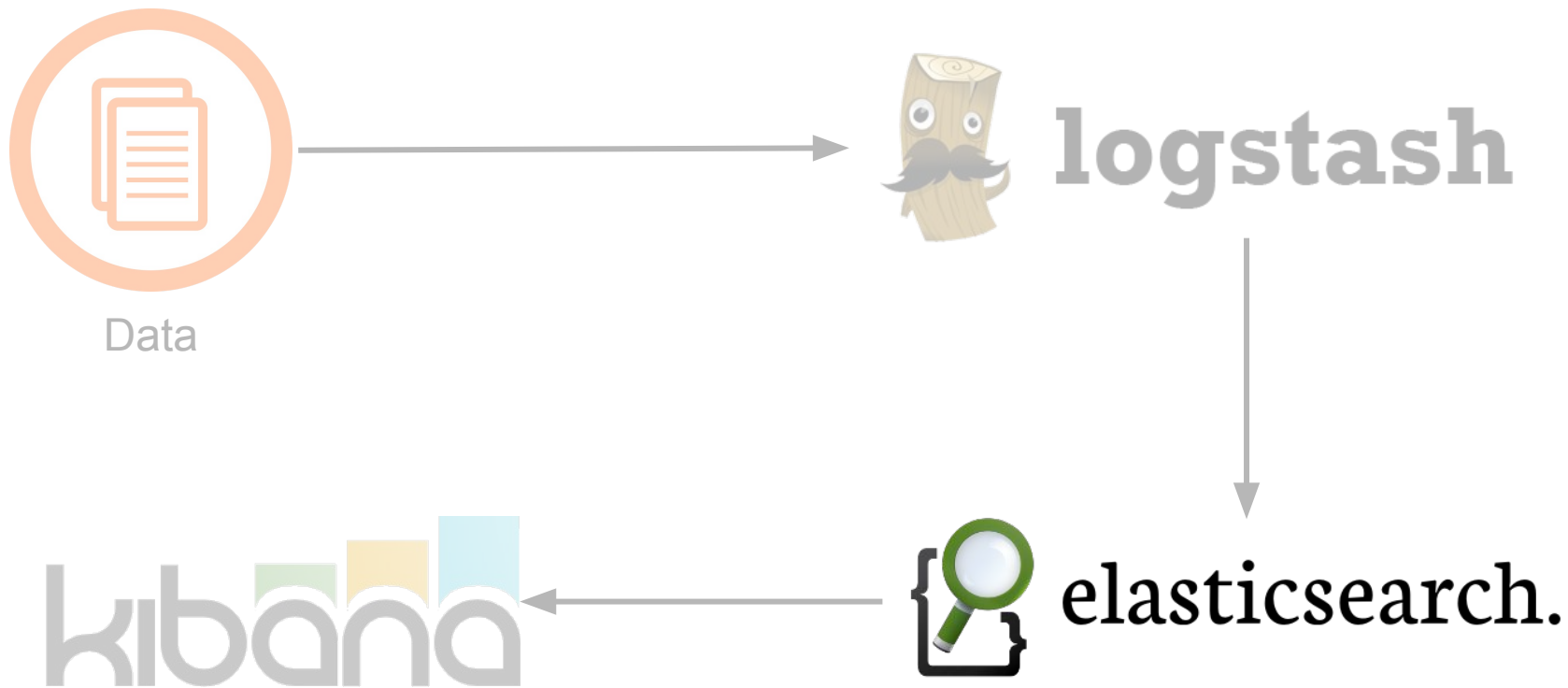
ip: 193.145.120.40
city: La Laguna
country: ES



Output

datastore
files
e-mail
pager
monitoring
chat
API
queues

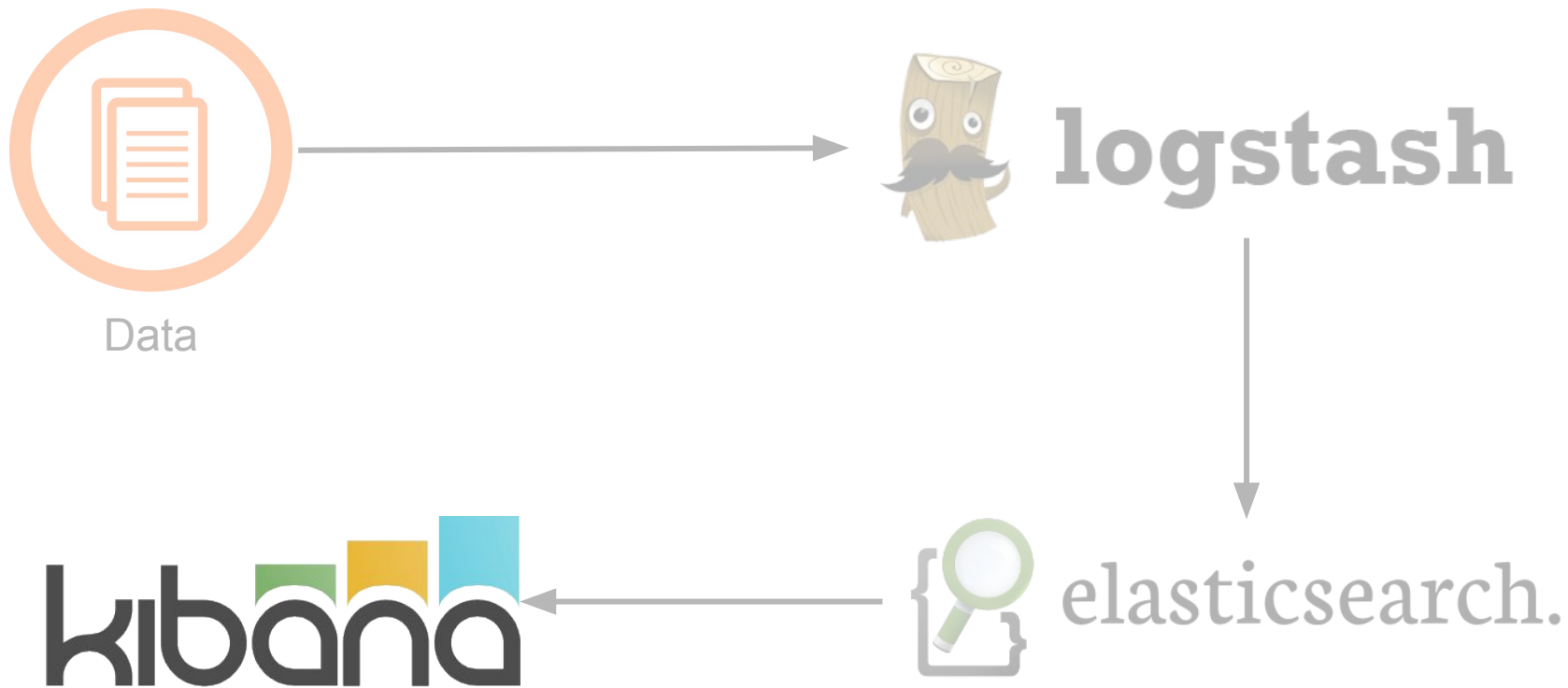
Flujo de datos



Elasticsearch

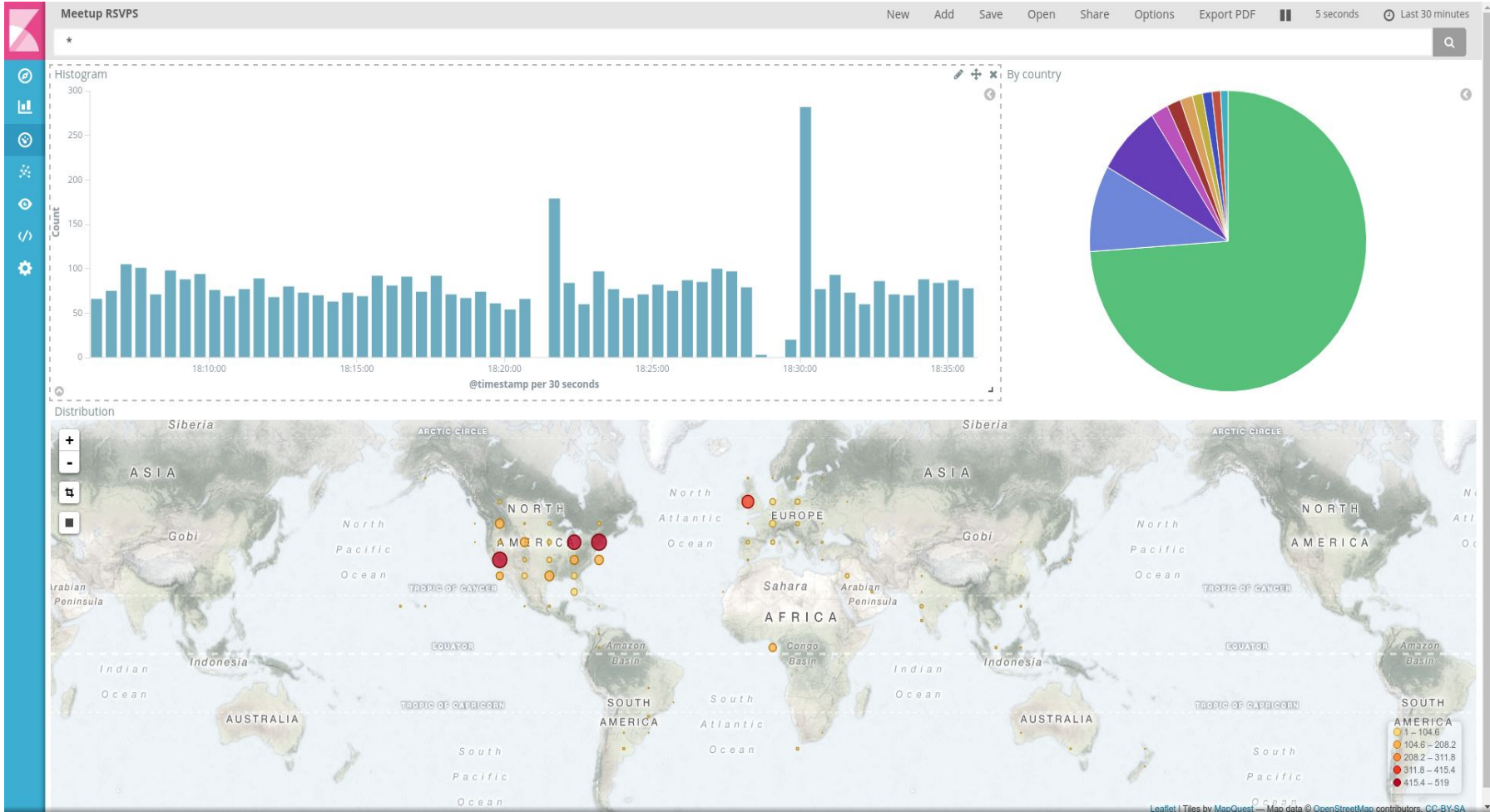
- Motor de búsqueda distribuido basado en REST y JSON
- Software libre: licencia Apache 2.0
- Lenguaje de búsquedas fácil de entender y muy potente:
 - Búsquedas de texto completo (frase, lógica difusa)
 - Búsqueda numérica (soporta rangos, fechas, direcciones IPv4)
 - Resaltado
 - Aggregations
 - Suggestions

Flujo de datos



Kibana

- Ejecuta consultas sobre los datos y muestra los resultados
- Permite trabajar con widgets
- Compartir/Guardar/Cargar dashboards
- Software libre: licencia Apache 2.0



Ejemplos ...

Iniciar el servicio de Elasticsearch

- `vagrant init adsaman/elk`
- `vagrant up --provider virtualbox`
- `vagrant ssh`
- `sudo service elasticsearch start`
- `sudo tail -f /var/log/elasticsearch/elasticsearch.log`

Primeras consultas...

- Comprobar si se está ejecutando ES:
 - `curl -XGET localhost:9200/`
- Estado del cluster:
 - `curl -XGET localhost:9200/_cat/health?v`
- Listar los índices:
 - `curl -XGET localhost:9200/_all/_settings?pretty`
 - `curl -XGET localhost:9200/_cat/indices?v`

Kibana

- Iniciar el servicio:
 - `sudo /etc/init.d/kibana start`
- Comprobar los logs:
 - `sudo tail -f /var/log/kibana/kibana.std*`
- Acceder a <http://localhost:5601/>
- En la versión 5 incorpora una consola (antiguo Sense)

X-Pack: Extension Pack for the Elastic Stack

- Paquete con varios plugins para ES/Kibana
- Incluye:
 - **Security:** ACL, encriptación, filtrado de IP, auditorías, etc.
 - **Monitoring:** informa del estado del cluster, índices, nodos, etc.
 - **Watcher:** alertas y notificaciones
 - **Reporting:** generación de informes
 - **Graph:** grafos que representan la relación entre diferentes términos
- <https://www.elastic.co/guide/en/x-pack/current/index.html>

URL de búsquedas

- Todos los elementos del índice meetups2, tipo meetup:
 - `curl "http://localhost:9200/meetups2/meetup/_search"`
- Respuestas a meetups en España :
 - `curl "/meetups2/_search?q=group.group_country:es"`
- Búsqueda de texto completo:
 - `curl "/meetups2/_search?q=museos&_source_include=event.event_name"`

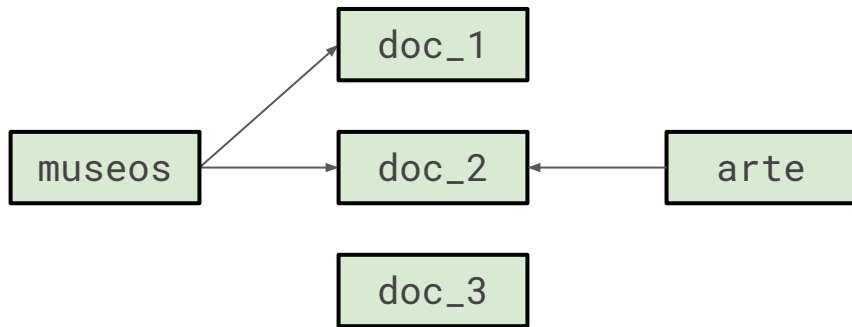
Search DSL

- Documentación:
 - <https://www.elastic.co/guide/en/elasticsearch/reference/master/query-dsl.html>

```
GET /meetups2/_search
{
  "query" : {
    "match" : {
      "group.group_country" : "es"
    }
  }
}
```


Inverted index

- ES utiliza índices “invertidos” para acelerar las búsquedas de texto completo
- Lista de todas las palabras apuntando a los documentos donde aparecen
- El contenido de los documentos se debe “tokenizar” y normalizar



Analyzer


```
GET /_analyze
{
  "analyzer": "standard",
  "text": "El caballo blanco de Santiago"
}
```

A diagram consisting of a horizontal arrow pointing from the input request box on the left to the output tokens box on the right. A vertical line segment extends downwards from the bottom of the input box, and another vertical line segment extends upwards from the top of the output box, meeting the horizontal arrow in the middle.

```
"tokens": [
  {
    "token": "el",
    "start_offset": 0,
    "end_offset": 2,
    "type": "<ALPHANUM>",
    "position": 0
  },
  {
    "token": "caballo",
    [...]
  },
  {
    "token": "blanco",
    [...]
  },
  {
    "token": "de",
    [...]
  },
  {
    "token": "santiago",
    [...]
  }
]
```

Analyzer

```
GET /_analyze
{
  "analyzer": "spanish",
  "text": "El caballo blanco de Santiago"
}
```

A horizontal arrow points from the right side of the input request box to the left side of the output response box, indicating the flow of data from the request to the response.

```
{
  "tokens": [
    {
      "token": "caball",
      "start_offset": 3,
      "end_offset": 10,
      "type": "<ALPHANUM>",
      "position": 1
    },
    {
      "token": "blanc",
      "start_offset": 11,
      "end_offset": 17,
      "type": "<ALPHANUM>",
      "position": 2
    },
    {
      "token": "santiag",
      "start_offset": 21,
      "end_offset": 29,
      "type": "<ALPHANUM>",
      "position": 4
    }
  ]
}
```

Reto

- Procesar los logs en <http://ddv.ull.es/users/asmarre/public/logs.tgz>
- Son 5,550,707 accesos a un servidor web Apache

Creación de un cluster de Elasticsearch

- Editamos el fichero de configuración `/etc/elasticsearch/elasticsearch.yml`

```
# ----- Discovery -----  
#  
# Pass an initial list of hosts to perform discovery when new node is started:  
# The default list of hosts is ["127.0.0.1", "[::1]"]  
#  
discovery.zen.ping.unicast.hosts: ["192.168.1.40", "192.168.1.41"]  
#  
# Prevent the "split brain" by configuring the majority of nodes (total number of nodes / 2  
# + 1):  
#  
discovery.zen.minimum_master_nodes: 1  
#  
# For more information, see the documentation at:  
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery.html>  
#
```

Configuración de Logstash

```
input {  
  stdin {}  
}  
  
filter {}  
  
output {  
  elasticsearch {  
    hosts => [ 'localhost:9200' ]  
    index => 'accesslog'  
  }  
  stdout { codec => rubydebug }  
}
```


Configuración de filtros para logs de Apache

```
filter {  
  grok {  
    match => { "message" => "%{COMBINEDAPACHELOG} \"%{HOSTNAME:http_host}\"" }  
  }  
  date {  
    match => [ "timestamp", "dd/MMM/yyyy:HH:mm:ss Z" ]  
  }  
  useragent {  
    source => "agent"  
  }  
  geoip {  
    source => "clientip"  
  }  
}
```

Grok

- Es un filtro de Logstash
- Procesa texto y genera datos estructurados
- Incluye una gran cantidad de patrones pre-definidos:
 - <https://github.com/logstash-plugins/logstash-patterns-core/tree/master/patterns>
- Documentación:
 - <https://www.elastic.co/guide/en/logstash/5.0/index.html>

Ejecutando logstash

```
$ alias logstash="sudo -u logstash /usr/share/logstash/bin/logstash --path.  
settings /etc/logstash"
```

```
$ cat /logs/access_15*.log | logstash -f /logs/logstash-apache-logs.conf  
Sending logstash logs to /var/log/logstash/logstash.log.  
Pipeline main started  
[...]
```

Otras cosas a tener en cuenta...

- Si no se ha definido previamente, ES infiere el tipo de datos de los campos
- En el caso del campo `geoip.location` se crea como un array de float
 - Pero realmente debería ser un `geo_point`
- No se pueden modificar los tipos de datos ya creados:
 - Hay que crear un nuevo índice y re-indexar todo el contenido
 - <https://www.elastic.co/blog/changing-mapping-with-zero-downtime>

Creación de un índice con un tipo asociado

```
PUT /apache-logs/
{
  "mappings": {
    "access": {
      "properties": {
        "geoip": {
          "properties": {
            "location": {
              "type": "geo_point"
            }
          }
        }
      }
    }
  }
}
```

Referencias

- Empresa detrás de todo este software:
 - <http://www.elastic.co>
- Documentación:
 - <https://www.elastic.co/guide/index.html>
- Libro “Elasticsearch: The Definitive Guide”:
 - <https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html>
- ELK en la nube:
 - <https://sematext.com/logsene/>

¿Preguntas?