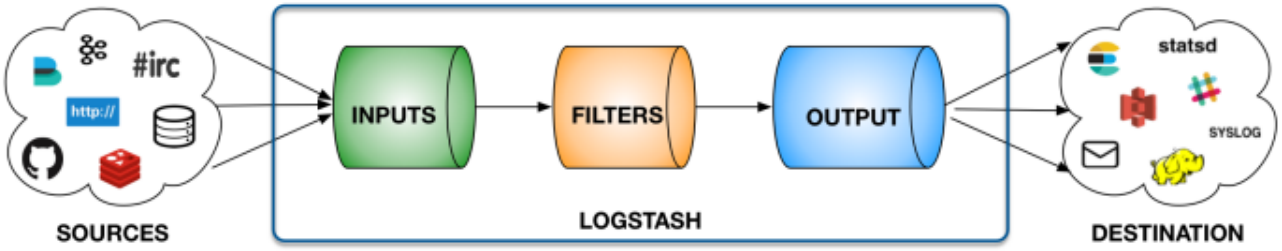
ELK: Logstask(procesamiento)

3.2_procesamiento.pdf



- /etc/logstash/conf.d/test1.conf

```
input {
  file {
    path => "/home/openweb/Documents/dataset/test.json"
    start_position => "beginning"
    codec => "json"
}}

output {
  stdout { codec => rubydebug }
}
```

input

- JSON
- https://www.elastic.co/guide/en/logstash/current/input-plugins.html

- lanzar logstash:`/usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/logstash.conf --path.settings=/etc/logstash`
- pasarle información: `echo '{ «name»:«John», «surname»:«Pitt»,«age»:«30», «cars»:[ «Ford», «BMW», «Fiat» ] }' » «/home/opweb/Documents/dataset/test.json»`

filter

- https://www.elastic.co/guide/en/logstash/current/filter-plugins.html
- mutate:

```
filter {
   mutate {
     remove_field => [ "@version" ]
     add_field => { "tipoUsuario" => "cliente" }
```

```
      gsub => ["surname", " - ", ""]
    }
}
```

- grok:

```
filter{
 grok {
   match => { "personalInformation" => [ "Information:%{WORD:Name}
%{WORD:Surname} %{NUMBER:age} %{NUMBER:height}" ] }
 }
}
```

    - ficheros sin formato «previo» o establecido.
- CIDR&GEOIP:

```
filter {
 if [srcip] and [srcip] != "N/A" {
   cidr {
     add_tag => ["src_ip_priv"]
     address => ["%{srcip}"]
     network =>
["172.16.0.0/12","10.0.0.0/8","192.168.0.0/16","169.254.0.0/16","0.0.0.0/32"]
   }
  if "src_ip_priv" not in [tags] {
    geoip {
      target => "src_geoip"
      source => "srcip"
      fields => ["city_name", "continent_code","country_code2",
"country_code3", "country_name","ip", "latitude", "longitude","location"]
    }
  }
 }
}
```

- instalación plugin: `/usr/share/logstash/bin/logstash-plugin install logstash-filter-cidr`

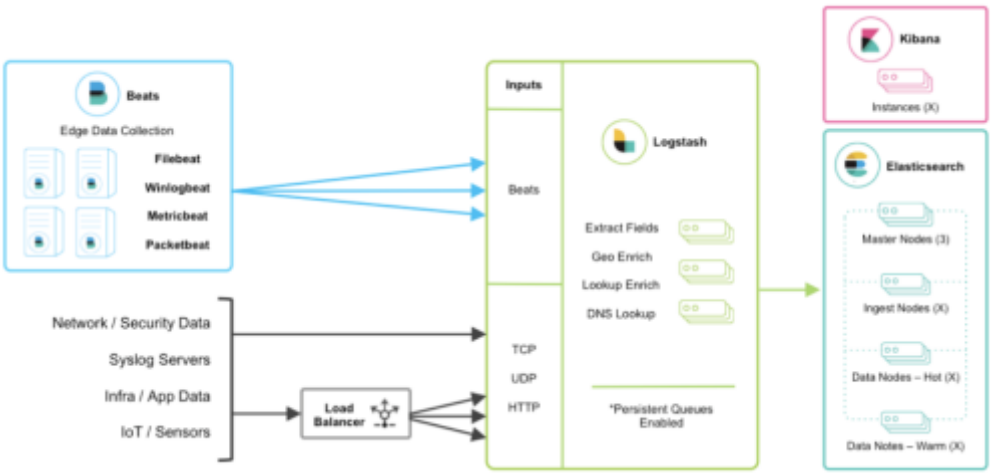
output

- https://www.elastic.co/guide/en/logstash/current/output-plugins.html

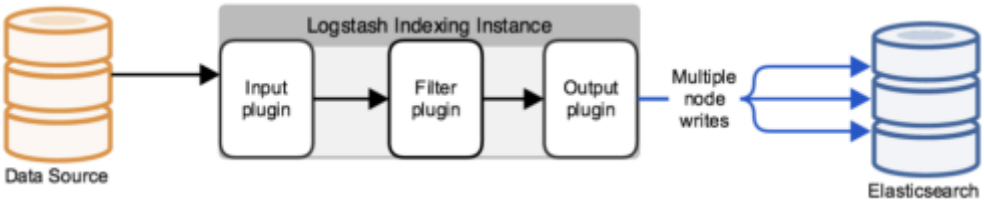
codecs

- otro filtro que puede operar como parte del input o output
- transporte ↔ formato
- populares:
    - JSON
    - Multiline: junta varias líneas para recoger un único evento (como excepciones en Java)
    - otros: avro,collectd,nmap,fluent,plain,s3_plain

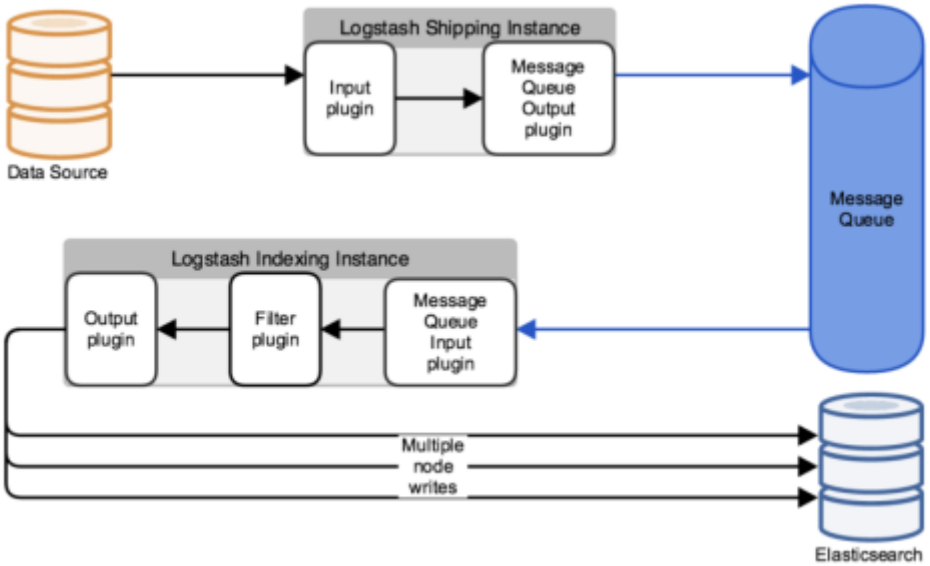
escalado



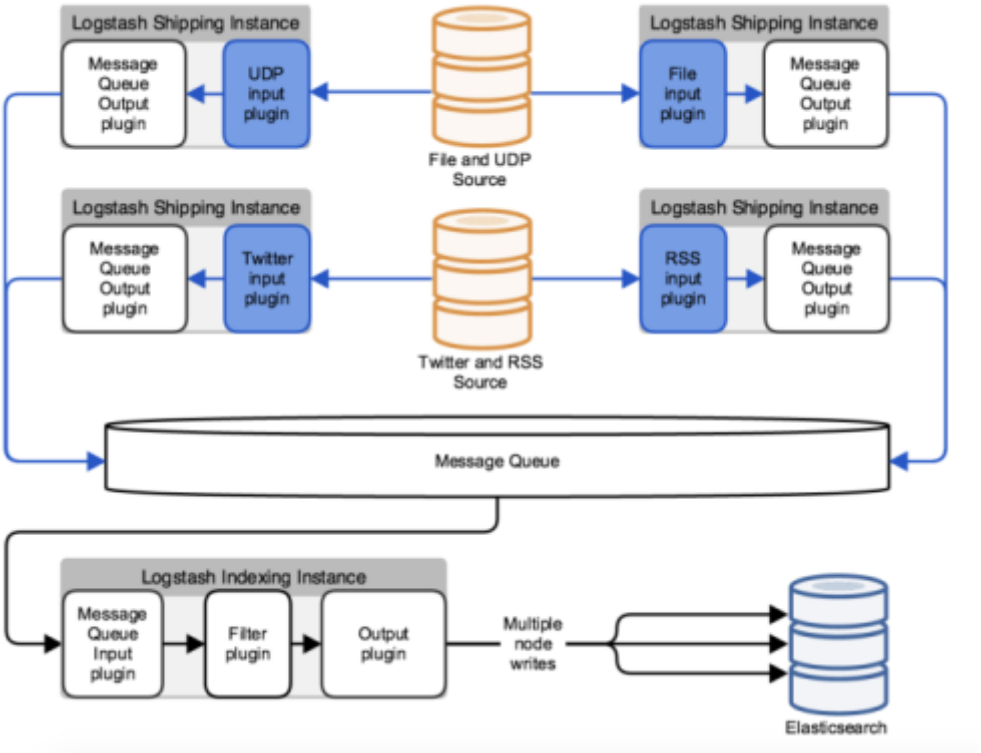
- procesado simple a cluster:



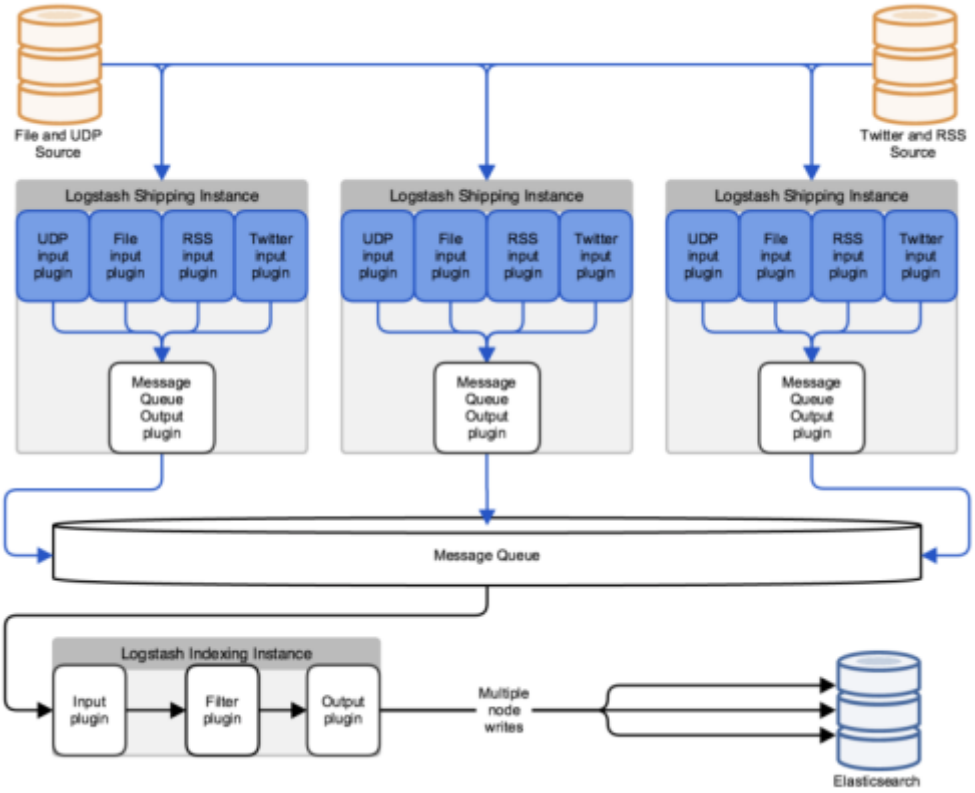
- buffer:



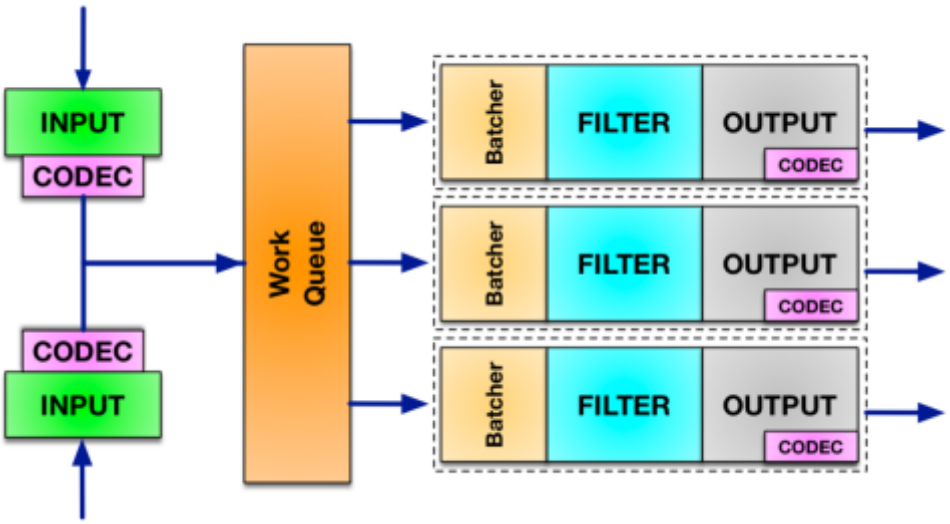
- Múltiples conexiones



- HA:



- multihilo:



- podemos decidir los cores que dedicamos a cada parte del proceso
- podemos dejarlo en auto y que se autogestione
- supervisar para ver si se produce cuello de botella en alguno de los puntos