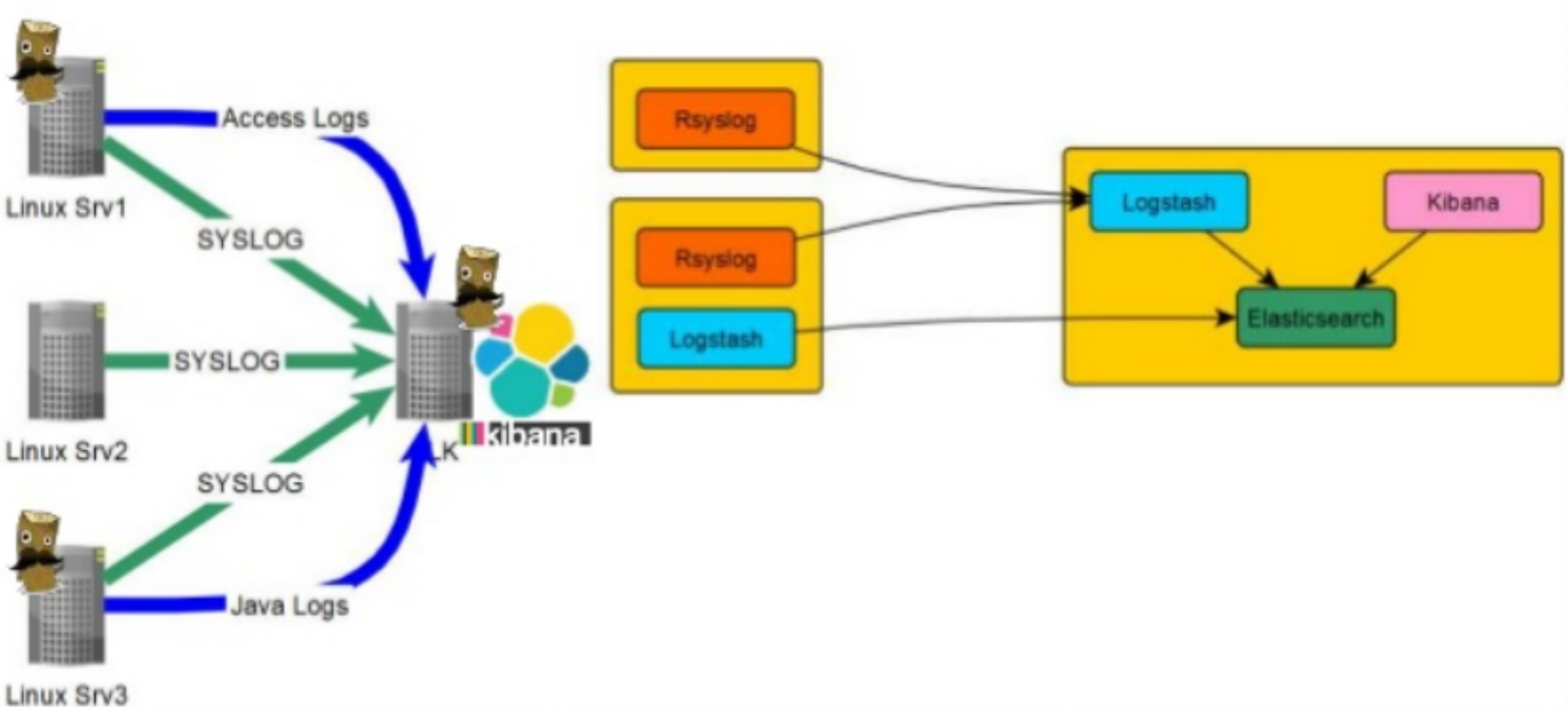
ELK : Elasticsearch + Logstash + Kibana



ELK : Elasticsearch + Logstash + Kibana

- Config Logstash très simple :

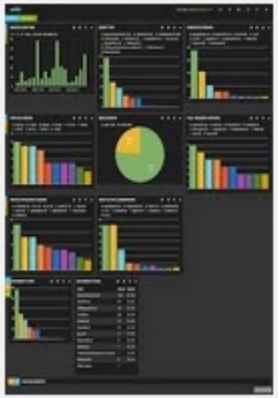
```
filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }
  date {
    match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss Z" ]
  }
}

output {
  elasticsearch { hosts => ["localhost:9200"] }
  ...
}
```

ELK : Elasticsearch + Logstash + Kibana

- Config Logstash très simple :

```
{
        "message" => "66.249.70.23 - - [14/Dec/2017:22:19:45
+0100] \"GET /simbad/sim-id?Ident=HD+++3674 HTTP/1.1\" 200 9609 \"-\"
"Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)\"",
     "@timestamp" => "14/Dec/2017:22:19:45Z",
       "@version" => "1",
       "clientip" => "66.249.70.23",
          "ident" => "-",
           "auth" => "-",
      "timestamp" => "14/Dec/2017:22:19:45 +0100",
           "verb" => "GET",
        "request" => "/simbad/sim-id?Ident=HD+++3674",
    "httpversion" => "1.1",
       "response" => "200",
          "bytes" => "9609",
       "referrer" => "-",
          "agent" => "\"Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)\""
}
```

ELK : Elasticsearch + Logstash + Kibana







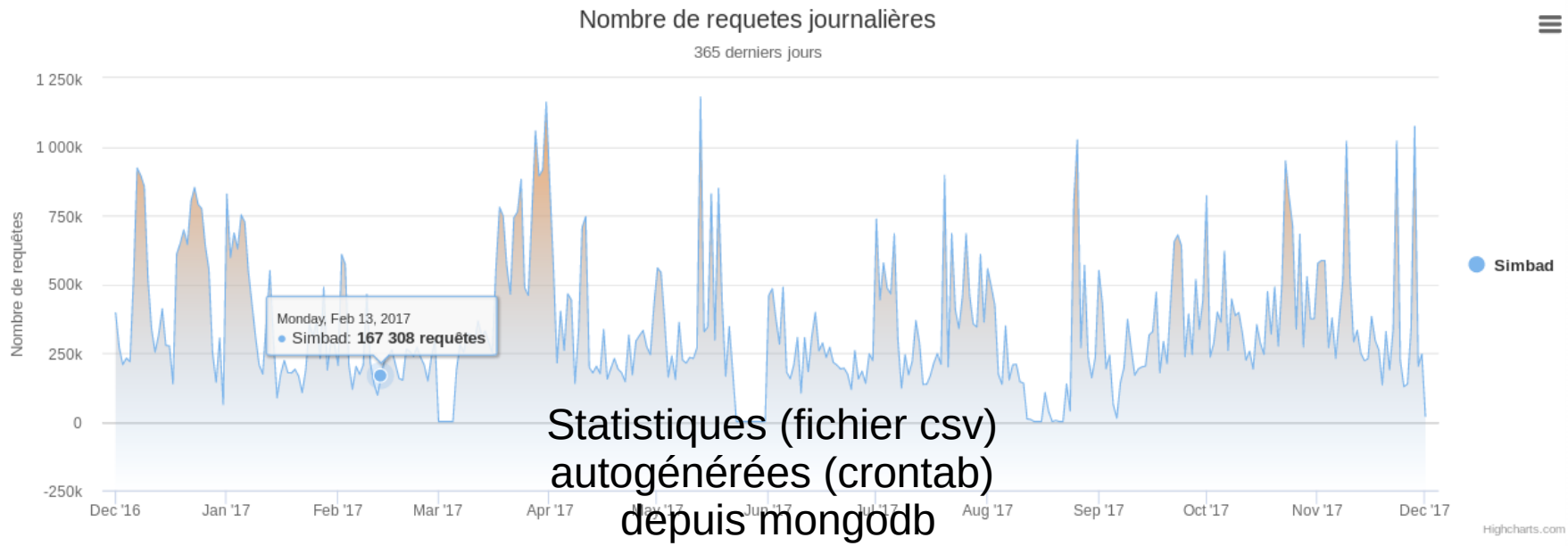


CDSLOG

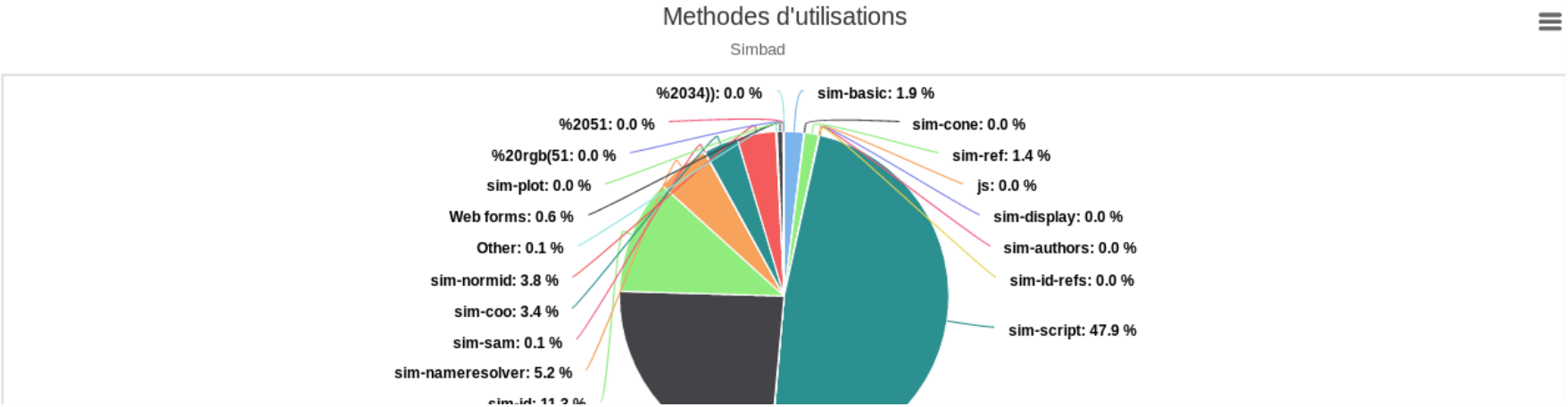
- cdslog : serveur de logs pour le CDS
- Crontab qui lit le répertoire /data1/simbad/http/ et lance un script qui insère le fichier json du jour dans la base mongodb
  - Dès que le fichier est lu, le nombre de lignes est stocké
  - Avant la lecture d'un fichier, on se décale au nombre de ligne précédemment lu
  - Fichier au format json


```
    "time":"00:00:00", "date":"2017-09-29",
    "ip_address":"130.79.128.30",
    "query_strings":{"id":"NLTT %2015885",
"data":"@,I.0,C.0,J,J.E,P,X,V,D,S,T,  %23B", "option":"strict"},
    "service":"Simbad",
    "user_agent":"wwwget\/3.14  (2014-06-25)",
    "method":"sim-nameresolver"
```

CDSLOG



Statistiques (fichier csv) autogénérées (crontab) depuis mongodb



CDSLOG : Site en Nodejs