Taller de ELK

De los ficheros de log al panel en una sesión

Grupos de Trabajo de RedIRIS
Universitat Politècnica de València, 15.11.2016

uc3m | Universidad Carlos III de Madrid

Nuria Prieto Pinedo / Rafael Calzada Pradas
Grupo de Seguridad

RedIRIS

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

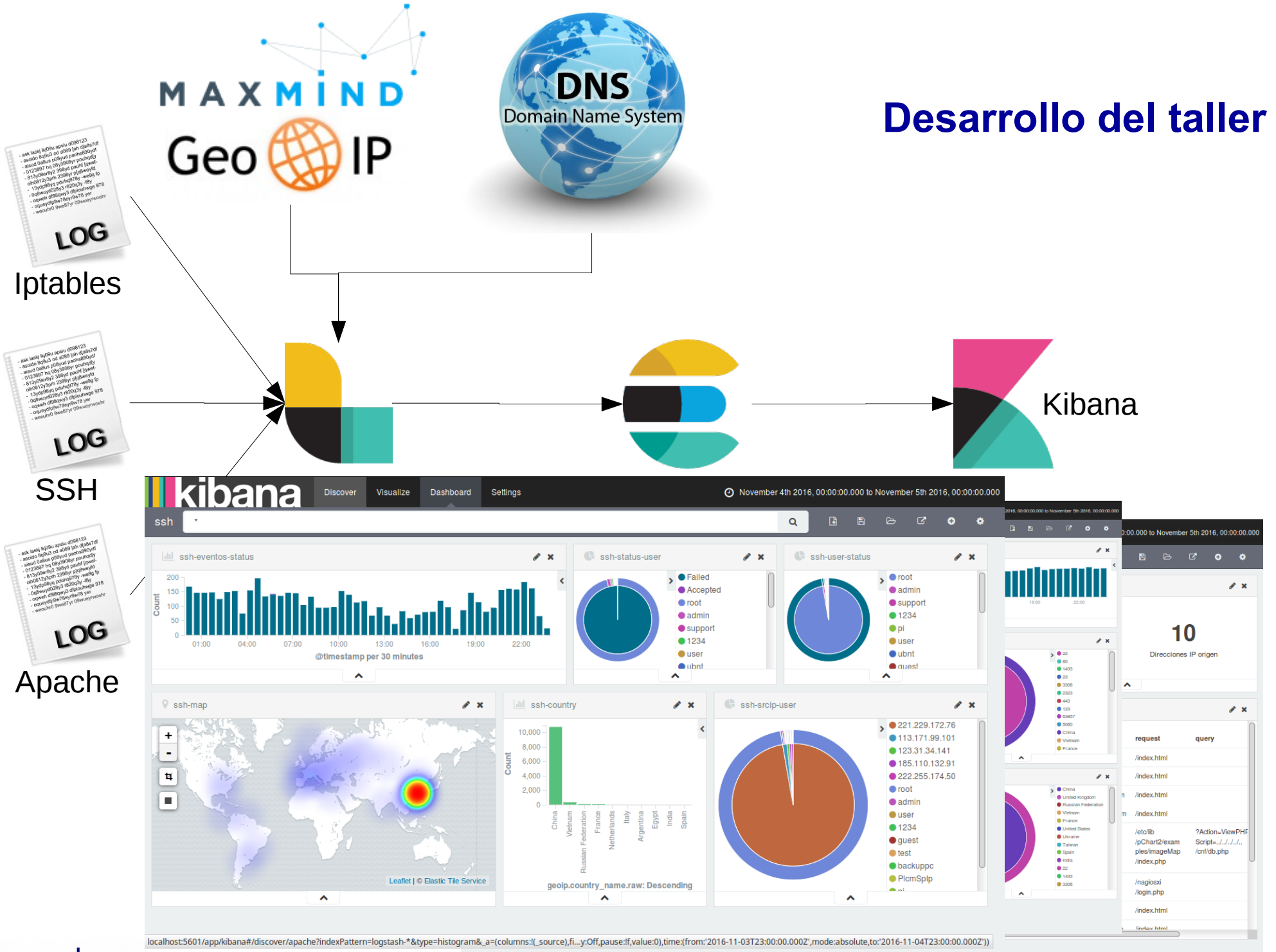
Material Necesario para el taller

- Sistema con VirtualBox instalado (o sistema compatible)
  - 20 GB de disco libres
  - 4 GB de RAM (2 GB para la máquina virtual)
  - Conexión a Internet
    - Para descargar algunos paquetes
    - Para consultar información
- Descargar la máquina virtual y ficheros de ejemplo:
  - http://c.rediris.es/jjtt2016-ELK
  - Usuario: iriselk
  - Contraseña: iriselk
- Datos basados en la idea de Fr4n, @fwhibbit
  - https://www.fwhibbit.es/24-horas-en-la-vida-de-mi-router-domestico

Presentar ELK basándonos en un caso práctico

- Instalación sobre Ubuntu 16.04 LTS
  - Logstash versión 2.4.0, Elasticsearch versión 2.4.1, Kibana 4.6.1
- Ingesta simple de eventos correspondientes a servicios habituales
  - iptables (syslog), SSH (syslog), Apache (json)
- Normalización y enriquecimiento de eventos
  - Resolución DNS, Geolocalización
- Monitorización del servicio Elasticsearch
  - API, Plugins opensource y comerciales
- Borrado de información
  - Índices completos, Borrado selectivo, Reindexación
- Visualización de la información
  - Histogramas, diagramas de tarta, Métricas, Mapas de geolocalización

Desarrollo del taller



Logstash

Motor de recolección de datos en tiempo real.

- Puede unificar los datos de fuentes distintas y normalizar dichos datos en el destino que se desee. Limpiando datos para poder realizar analítica y visualización de datos

Tres etapas en el tratamiento de la información:

- INPUT:Recoge la información de los eventos y hace un análisis básico
- FILTER: Analiza la información del evento. Es opcional y se puede paralelizar.
  - Asigna valores a etiquetas
  - Obtiene información complementaria:
  - Resolución DNS
  - Geolocalización
  - Modificación de valores
- OUTPUT: Entrega la información procesada

Instalación y configuración básica de logstash

Iniciamos sesión con el usuario `iriselk/iriselk`

```
# dpkg -i logstash-2.4.0_all.deb
```

Creamos el fichero `/etc/logstash/input/input-syslog.conf` con el siguiente contenido

```
input {
        stdin{ type => “syslog” }
}

filter { }

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

Primera prueba

- Vamos a pasar dos eventos por logstash con la configuración básica

```
# head -2 /home/iriselk/logs/messages-4Nov.txt |/opt/logstash/bin/logstash -f
/etc/logstash/input/input-syslog.conf
Settings: Default pipeline workers: 1
Pipeline main started
{
       "message" => "Nov  4 00:00:00 honeeepi kernel: [308877.538952] IN=eth0 OUT=
MAC=b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:4c:bd:81:40:00:2e:06:43:80
SRC=221.229.172.74 DST=192.168.0.210 LEN=76 TOS=0x00 PREC=0x00 TTL=46 ID=48513 DF
PROTO=TCP SPT=28314 DPT=22 WINDOW=229 RES=0x00 ACK PSH URGP=0 ",
      "@version" => "1",
    "@timestamp" => "2016-11-13T22:42:47.524Z",
          "type" => "syslog",
          "host" => "iriselk"
}
{
       "message" => "Nov  4 00:00:00 honeeepi kernel: [308877.542089] IN=eth0 OUT=
MAC=b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:34:bd:82:40:00:2e:06:43:97
SRC=221.229.172.74 DST=192.168.0.210 LEN=52 TOS=0x00 PREC=0x00 TTL=46 ID=48514 DF
PROTO=TCP SPT=28314 DPT=22 WINDOW=229 RES=0x00 ACK FIN URGP=0 ",
      "@version" => "1",
    "@timestamp" => "2016-11-13T22:42:47.864Z",
          "type" => "syslog",
          "host" => "iriselk"
}
```

Formato del fichero de entrada: iptables

```
Nov  4 00:00:00 honeeepi kernel: [308877.538952] IN=eth0 OUT=
MAC=b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:4c:bd:81
:40:00:2e:06:43:80 SRC=221.229.172.74 DST=192.168.0.210 LEN=76
TOS=0x00 PREC=0x00 TTL=46 ID=48513 DF PROTO=TCP SPT=28314
DPT=22 WINDOW=229 RES=0x00 ACK PSH URGP=0
```

```
Nov  4 00:00:00 honeeepi kernel: [308877.542089] IN=eth0 OUT=
MAC=b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:34:bd:82
:40:00:2e:06:43:97 SRC=221.229.172.74 DST=192.168.0.210 LEN=52
TOS=0x00 PREC=0x00 TTL=46 ID=48514 DF PROTO=TCP SPT=28314
DPT=22 WINDOW=229 RES=0x00 ACK FIN URGP=0
```

Análisis de texto

- Utilizamos el filtro `grok`
- https://www.elastic.co/guide/en/logstash/2.4/plugins-filters-grok.html
- Probamos el patrón en https://grokdebug.herokuapp.com/
- Pistas:
  - Utilizar patrones predefinidos %{PATRON:ETIQUETA}
    - Información en https://github.com/logstash-plugins/logstash-patterns-core/tree/master/patterns
    - Fecha y Hora: `%{SYSLOGTIMESTAMP:timestamp}`
    - Nombre de máquina: `%{HOSTNAME:_host}`
    - Dirección IP: `%{IP:src_ip}`
  - Expresiones regulares (?<ETIQUETA>EXPRESION_REGULAR)
    - Información en https://github.com/kkos/oniguruma/blob/master/doc/RE
    - `MAC=(?<mac_addr>\S+)`
  - `También vale buscar en google`

- Guardamos el patrón en `/etc/logstash/patterns/iriselk`

```
IPTABLES %{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:loghostname} kernel: \
[\d+\.\d+\] IN=(?<in>[a-z]+[0-9]+) OUT= MAC=(?<mac_addr>\S+) SRC=%{IP:src_ip}
DST=%{IP:dst_ip} LEN=\d+ TOS=0x\d+ PREC=0x\d+ TTL=\d+ ID=\d+(?:\sDF)? PROTO=(?
<proto>\S+) SPT=(?<src_port>\d+) DPT=(?<dst_port>\d+)(?:\sWINDOW=\d+)?
(?:\sRES=0x\d+)?(?:\s[ACKSYNFIRT]{3})?(?:\sURGP=\d)?(?:\sLEN=\d+)?
```

- Modificamos la etapa `filter`

```
filter {
        grok {
                patterns_dir => "/etc/logstash/patterns"
                match => [ "message", "%{IPTABLES}" ]
                add_tag => [ "grokdked_iptables" ]
                tag_on_failure => ["_grokparsefailure"]
                remove_tag => [ "_grokparsefailure" ]
                keep_empty_captures => false
        }
}
```

Resultado

```
# head -2 /home/iriselk/logs/messages-4Nov.txt |/opt/logstash/bin/logstash -f
/etc/logstash/input/input-syslog.conf
Settings: Default pipeline workers: 1
Pipeline main started
{
        "message" => "Nov  4 00:00:00 honeeepi kernel: [308877.538952] IN=eth0 OUT=
MAC=b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:4c:bd:81:40:00:2e:06:43:80
SRC=221.229.172.74 DST=192.168.0.210 LEN=76 TOS=0x00 PREC=0x00 TTL=46 ID=48513 DF
PROTO=TCP SPT=28314 DPT=22 WINDOW=229 RES=0x00 ACK PSH URGP=0 ",
       "@version" => "1",
     "@timestamp" => "2016-11-15T16:27:34.284Z",
           "type" => "syslog",
           "host" => "iriselk",
      "timestamp" => "Nov  4 00:00:00",
    "loghostname" => "honeeepi",
             "in" => "eth0",
       "mac_addr" =>
"b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:4c:bd:81:40:00:2e:06:43:80",
         "src_ip" => "221.229.172.74",
         "dst_ip" => "192.168.0.210",
          "proto" => "TCP",
       "src_port" => "28314",
       "dst_port" => "22",
           "tags" => [
        [0] "grokdked_iptables"
    ]
}
```

¿Por qué son distintos?

Modificando etiquetas y valores

- Vamos a utilizar el filtro `mutate`
  - https://www.elastic.co/guide/en/logstash/2.4/plugins-filters-mutate.html
  - Para convertir el formato de `timestamp` a `string`
  - Para eliminar una etiqueta que ya no necesitamos (si hemos podido convertir la fecha)
- Y el filtro `date`
  - https://www.elastic.co/guide/en/logstash/2.4/plugins-filters-date.html
  - Para analizar la fecha

Utilizando la fecha contenida en el mensaje de syslog

```
mutate {
        convert => [ "timestamp", "string" ]
}
date {
        match => ["timestamp",
                  "MMM dd HH:mm:ss",
                  "MMM  d HH:mm:ss" ]
}

if "_dateparsefailure" not in ["tags"] {
        mutate {
                remove_field => [ "timestamp" ]
        }
}
```

Evento con el campo @timestamp corregido

```
{
        "message" => "Nov  4 00:00:00 honeeepi kernel: [308877.542089]
IN=eth0 OUT=
MAC=b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:34:bd:82:40:00:2e:06:
43:97 SRC=221.229.172.74 DST=192.168.0.210 LEN=52 TOS=0x00 PREC=0x00 TTL=46
ID=48514 DF PROTO=TCP SPT=28314 DPT=22 WINDOW=229 RES=0x00 ACK FIN URGP=0 ",
       "@version" => "1",
     "@timestamp" => "2016-11-14T00:46:10.720Z",
           "type" => "syslog",
           "host" => "iriselk",
      "timestamp" => "Nov  4 00:00:00",
    "loghostname" => "honeeepi",
             "in" => "eth0",
       "mac_addr" =>
"b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:34:bd:82:40:00:2e:06:43:
97",
         "src_ip" => "221.229.172.74",
         "dst_ip" => "192.168.0.210",
          "proto" => "TCP",
       "src_port" => "28314",
       "dst_port" => "22",
           "tags" => [
        [0] "grokdked_iptables"
    ]
}
```

Enriqueciendo eventos: DNS

- Utilizaremos el filtro `dns` y `mutate`
  - Para resolver al nombre DNS de `src_ip`
  - Para crear una copia de la etiqueta `src_ip` a `src_hostname`
- Información
  - https://www.elastic.co/guide/en/logstash/2.4/plugins-filters-dns.html

```
filter {
  dns {
    Nameserver => [ “servidor1”, “servidor2”]
    reverse => [ "source_host", "field_with_address" ]
    resolve => [ "field_with_fqdn" ]
    action => "replace"
  }
}
```

Solución al enriquecimiento DNS

```
mutate {
      add_field => { "src_hostname" => "%{src_ip}" }
}

dns {
      nameserver => ["8.8.8.8"]
      failed_cache_size => "1000"
      hit_cache_size => "1000"
      hit_cache_ttl => "300"
      reverse => [ "src_hostname" ]
      action => "replace"
}

# Si la resolución no ha sido posible, borramos la etiqueta
if [src_hostname] == [src_ip] {
      mutate {
            remove_field => [ "src_hostname" ]
      }
}
```

Resultado de enriquecimiento DNS

- No todas las IPs están registradas en DNS….

```
# grep "74.82.47.34" /home/iriselk/logs/messages-4Nov.txt |head -2
|/opt/logstash/bin/logstash -f input-syslog.conf
Settings: Default pipeline workers: 1
Pipeline main started
{
         "message" => "Nov  4 00:01:56 honeeepi kernel: [308993.108367] IN=eth0 OUT=
MAC=b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:28:d4:31:00:00:ed:06:be:af
SRC=74.82.47.34 DST=192.168.0.210 LEN=40 TOS=0x00 PREC=0x00 TTL=237 ID=54321 PROTO=TCP
SPT=41376 DPT=9200 WINDOW=65535 RES=0x00 SYN URGP=0 ",
        "@version" => "1",
      "@timestamp" => "2016-11-03T23:01:56.000Z",
            "type" => "syslog",
            "host" => "iriselk",
     "loghostname" => "honeeepi",
              "in" => "eth0",
         "mac_addr" =>
"b8:27:eb:99:b8:62:54:67:51:09:86:e7:08:00:45:00:00:28:d4:31:00:00:ed:06:be:af",
          "src_ip" => "74.82.47.34",
          "dst_ip" => "192.168.0.210",
           "proto" => "TCP",
        "src_port" => "41376",
        "dst_port" => "9200",
            "tags" => [
        [0] "grokdked_iptables"
    ],
    "src_hostname" => "scan-09h.shadowserver.org"
}
```

Enriquecimiento de eventos con Geolocalización

- Es necesario instalar el paquete `geoip-database-contrib`
  - Descarga la base de datos de geolocalización gratuita de Maxmind
    - `# apt-get install geoip-database-contrib`
- Se utiliza el filtro `geoip`
  - https://www.elastic.co/guide/en/logstash/2.4/plugins-filters-geoip.html

```
geoip {
        source => "src_ip"
        target => "geoip"
        database => "/usr/share/GeoIP/GeoLiteCity.dat"
        add_field => [ "[geoip][coordinates]",
                       "%{[geoip][longitude]}" ]
        add_field => [ "[geoip][coordinates]",
                       "%{[geoip][latitude]}" ]
}
```

Resultado de enriquecimiento con Geolocalización

```
          "geoip" => {
                        "ip" => "221.229.172.74",
           "country_code2" => "CN",
           "country_code3" => "CHN",
            "country_name" => "China",
          "continent_code" => "AS",
             "region_name" => "04",
               "city_name" => "Nanjing",
                "latitude" => 32.0617,
               "longitude" => 118.77780000000001,
                "timezone" => "Asia/Shanghai",
       "real_region_name" => "Jiangsu",
                "location" => [
           [0] 118.77780000000001,
           [1] 32.0617
       ]
   }
}
```

Elasticsearch

Motor de búsqueda y analítica escalable

Permite almacenar, buscar y analizar grandes volúmenes de datos casi en tiempo real.

Generalmente se utiliza como tecnología subyacente en aplicaciones con búsquedas complejas.

Ejemplo:

- Para recolectar los logs o transacciones que se desean analizar y extraer información para detectar tendencias, resúmenes, o anomalías. En este caso, se utiliza Logstash para recolectar, agregar y normalizar los datos. Luego Logstash inserta estos datos en Elasticsearch. Una vez que los datos están en Elasticsearch, pueden realizarse búsquedas y agregaciones para extraer la información que se considere de interés.

Instalación y configuración de elasticsearch

- Iniciamos sesión con el usuario `iriselk/iriselk`
- Instalación de Elasticsearch

```
# dpkg -i elasticsearch-2.4.1.deb
```

- Configuración:
  - Modificaremos la cantidad de memoria dedicada a la pila (heap) de Java, que debe ser inferior al 50% de la RAM disponible, en nuestro caso hemos puesto 600m. No debe superarse los 32g
    - `/etc/default/elasticsearch`
    - `ES_HEAP_SIZE=600m`
- Parámetros relacionados con el nodo y cluster de Elasticsearch.
  - Para este taller no modificaremos nada.
    - `/etc/elasticsearch/elasticsearch.yml`
- Ahora ya podemos lanzar el servicio de Elasticsearch

```
# service elasticsearch start
```

Inserción de eventos en Elasticsearch

- Para el desarrollo del taller, vamos a deshabilitar la resolución DNS
  - Se realiza en un único hilo de ejecución, y salvo que tengamos un servidor DNS local, ralentiza mucho la ejecución.
- Modificamos la etapa output, utilizaremos el plugin elasticsearch
  - https://www.elastic.co/guide/en/logstash/2.4/plugins-outputs-elasticsearch.html

```
output {
     elasticsearch { }
}
```

- Nos valen los valores por defecto:
  - host: `["127.0.0.1"]`
  - index: `"logstash-%{+YYYY.MM.dd}"`

Insertamos los eventos de iptables

- Tardará unos 10-15 minutos en función del equipo la memoria que hayamos asignado a la VM

```
# cat /home/iriselk/logs/messages-4Nov.txt | /opt/logstash/bin/logstash -f
/etc/logstash/input/input-syslog.conf
```

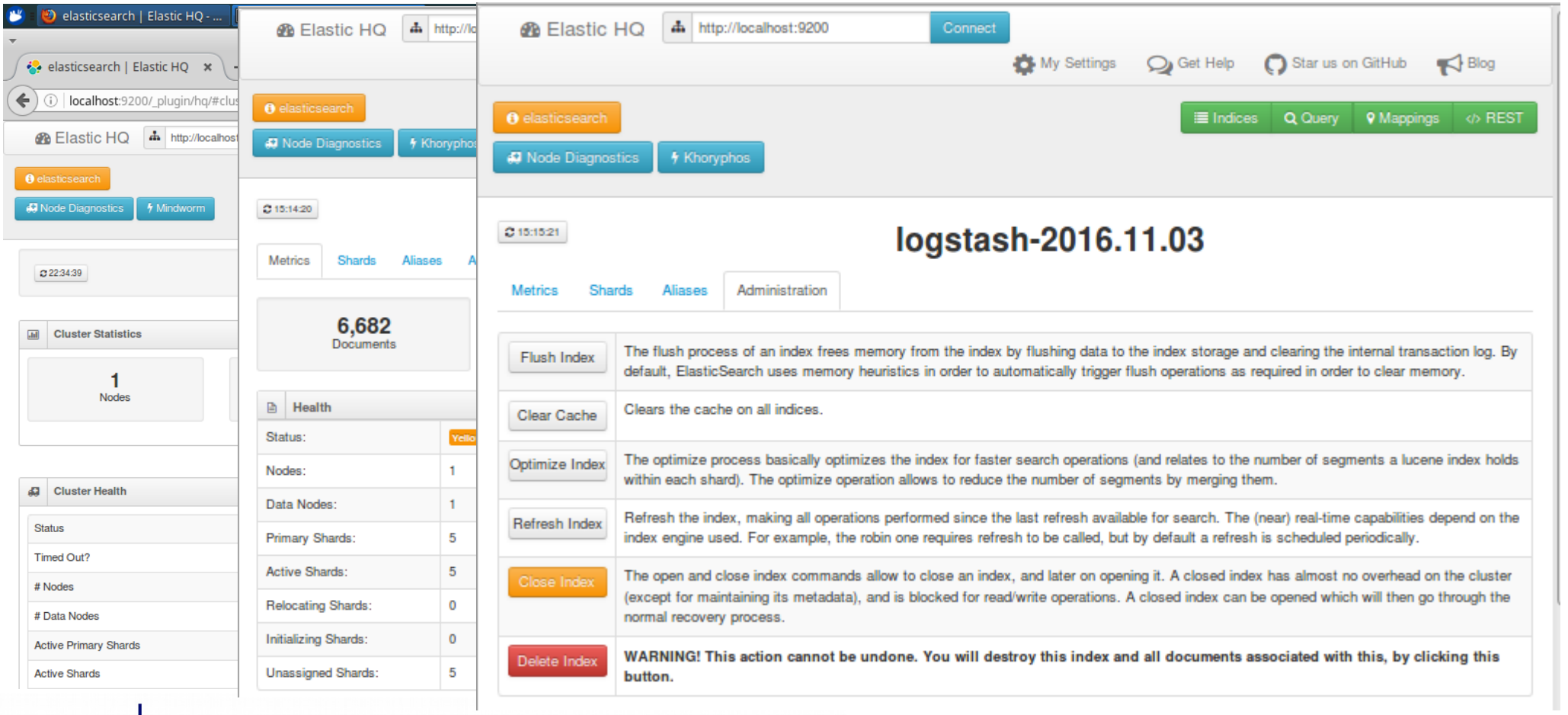
- Mientras tanto vamos a:
  - Ver como funciona nuestro Elasticsearch
  - Tratar los eventos de SSH y Apache

Conociendo el estado de nuestro Elasticsearch: API

- No apropiado para un primer momento
  - Información sobre todos los aspectos
    - `http://localhost:9200/_cat`
    - https://www.elastic.co/guide/en/elasticsearch/reference/2.4/indices-stats.html
  - Estado del cluster
    - `http://localhost:9200/_cluster`
    - https://www.elastic.co/guide/en/elasticsearch/reference/2.4/cluster-health.htm l
  - Estado de los nodos
    - `http://localhost:9200/_nodes`
    - https://www.elastic.co/guide/en/elasticsearch/reference/2.4/cluster-nodes-stats.html
  - Estado de los índices
    - `http://localhost:9200/_stats`
    - https://www.elastic.co/guide/en/elasticsearch/reference/2.4/indices-stats.html

Conociendo el estado de nuestro Elasticsearch: Plugin HQ

- Plugin de dominio público
- Instalación como la de cualquier plugin de Elasticsearch
  - `/usr/share/elasticsearch/bin/plugin install royrusso/elasticsearch-HQ`



Conociendo el estado de nuestro Elasticsearch: Otros plugins

- Otros plugins opensource
  - BigDesk: http://bigdesk.org/
  - Elasticsearch-Head: http://mobz.github.io/elasticsearch-head/
  - Kopf: https://github.com/lmenezes/elasticsearch-kopf
- Plugins comerciales
  - "X-Pack" de Elastic, que incluye entre otros el módulo Marvel.
    - En su versión básica, permite la monitorización de Elasticsearch, más información en https://www.elastic.co/subscriptions

Borrado de Datos

- Borrado de índices completos
  - Plugin HQ
  - API, https://www.elastic.co/guide/en/elasticsearch/reference/2.4/indices-delete-index.html

```
$ curl -XDELETE 'http://localhost:9200/logstash-2016.11.15/'
```

- Borrado selectivo de datos, Plugin `delete-by-query`
  - https://www.elastic.co/guide/en/elasticsearch/plugins/2.4/plugins-delete-by-query.html
  - https://www.elastic.co/guide/en/elasticsearch/reference/2.4/query-dsl.html
- Reindexación, para cuando se se compactan los índices
  - https://www.elastic.co/guide/en/elasticsearch/reference/2.4/docs-reindex.html

Muestra de eventos SSH

```
Nov  4 05:49:01 raspberrypi sshd[6669]: Accepted password for loreal from 192.168.0.6 port 39368 ssh2

Nov  4 05:49:01 raspberrypi sshd[6669]: pam_unix(sshd:session): session opened for user loreal by (uid=0)

Nov  4 05:49:02 raspberrypi sshd[6675]: Received disconnect from 192.168.0.6: 11: disconnected by user

Nov  4 05:49:02 raspberrypi sshd[6669]: pam_unix(sshd:session): session closed for user loreal

Nov  4 05:49:07 raspberrypi sshd[6681]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0
tty=ssh ruser= rhost=221.229.172.76  user=root

Nov  4 05:49:09 raspberrypi sshd[6681]: Failed password for root from 221.229.172.76 port 36548 ssh2

Nov  4 05:49:15 raspberrypi sshd[6681]: Received disconnect from 221.229.172.76: 11:  [preauth]

Nov  4 05:49:15 raspberrypi sshd[6681]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh
ruser= rhost=221.229.172.76  user=root
```

Aqui ya empieza a ser necesario utilizar referencias...

Patrones de análisis de logs SSH

```
SSH_INICIO (?<status>Accepted|Failed|Invalid user) (?<method>.*)
for (?<invalid>(invalid user )*)(?<user>\w+) from (?<src_ip>.*)
port (?<src_port>\d+) (?<version>.*)

SSH_FIN .*Received disconnect from (?<src_ip>[\d\.]*)

SSH_FIN2 .*Connection closed by (?<src_ip>[\d\.]*)

SSH_FIN3 .*session (closed|opened) for user (?<user>\w+)

SSH_PAM_SESSION1 pam_unix\(sshd:auth\): authentication failure;
logname=(?<logname>\w*) uid=(?<uid>\w+).*rhost=(?<src_ip>[\d.]
+)\s+user=(?<user>\w+)

SSH_PAM_SESSION2 PAM 2 more authentication failures; logname=(?
<logname>\w*) uid=(?<uid>\w+).*rhost=(?<src_ip>[\d.]+)\s+user=(?
<user>\w+)

SSH_PAM_SESSION (%{SSH_PAM_SESSION1}|%{SSH_PAM_SESSION2})

SSH %{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:_host} sshd\[\d+\]: (%
{SSH_FIN}|%{SSH_FIN2}|%{SSH_FIN3}|%{SSH_INICIO}|%{SSH_PAM_SESSION})
```

Fichero de configuración de logstash para ssh

Prácticamente igual que el de Iptables (lo salvamos como **`/etc/logstash/filter/input-ssh.conf`**)

```
input {
      stdin { type => "syslog "}
}

filter {
grok {
    patterns_dir => "/etc/logstash/patterns"
    match => [ "message", "%{SSH}" ]
    add_tag => [ "grokdked_ssh" ]
    tag_on_failure => ["_grokparsefailure"]
    remove_tag => [ "_grokparsefailure" ]
    keep_empty_captures => false
}
```

Y vamos insertando los eventos en Elasticsearch de la misma forma

```
# cat /home/iriselk/logs/auth.log | /opt/logstash/bin/logstash -f
/etc/logstash/input/input-ssh.conf
```

Configuración de Apache para generar logs en formato JSON

- Se ha seguido la configuración mostrada en:
  https://blog.logentries.com/2014/08/json-logging-in-apache-and-nginx-with-logentries/

- Muestras de logs

```
{    "time":"[04Nov/2016:04:13:05 +0000]",
 "remoteIP":"168.1.128.74",
     "host":"127.0.1.1",
  "request":"/index.html",
    "query":"",
   "method":"GET",
   "status":"200",
"userAgent":"Mozilla/5.0 (Windows NT 6.1; rv:31.0) Gecko/20100101 Firefox/31.0",
  "referer":"-" }
```

- **No tenemos que analizar nada…**
  - Usar el filtro `json`
    - https://www.elastic.co/guide/en/logstash/2.4/plugins-filters-json.html
  - Luego `mutate` para cambiar las etiquetas `time` y `remoteIP`

Fichero de configuración para logs en formato JSON

```
input {
   stdin { type => "syslog" }
}

filter {
   json {
         source => "message"
   }

   mutate {
         rename => { "time" => "timestamp" }
            rename => { "remoteIP" => "src_ip" }
                  add_tag => [ "grooked_apache" ]
   }
   mutate {
         convert => [ "timestamp", "string" ]
   }
   date {
            match => ["timestamp", "ISO8601", "[dd/MMM/YYYY:HH:mm:ss Z]" ]
   }

   if "_dateparsefailure" not in ["tags"] {
        mutate {
                  remove_field => [ "timestamp" ]
        }
   }
```

Y vamos insertando los eventos en Elasticsearch de la misma forma

```
# cat /home/iriselk/logs/access.log | /opt/logstash/bin/logstash -f
/etc/logstash/input/input-json.conf
```

- Plataforma de analítica y visualización diseñada para trabajar con Elasticsearch.
- Se utiliza para buscar, ver e interactuar con los datos almacenados en los índices de Elasticsearch.
- Pueden realizarse análisis y visualizaciones avanzadas de datos mediante una gran variedad de diagramas, tablas y mapas.

Interfaz web simple, que permite la creación de paneles para visualizar los contenidos de Elasticsearch en tiempo real.

Por eso, generalmente hablaremos de KIBANA, aunque nos estaremos refiriendo al stack ELK

Kibana: Instalación y Configuración

- Iniciamos sesión con el usuario `iriselk/iriselk`
- Instalación de Kibana

```
# dpkg -i kibana-4.6.1-amd64.deb
```

- Configuración `/opt/kibana/config/kibana.yml`

```
# Time in milliseconds to wait for responses from the back end or
elasticsearch.

# This must be > 0

elasticsearch.requestTimeout: 300000
```

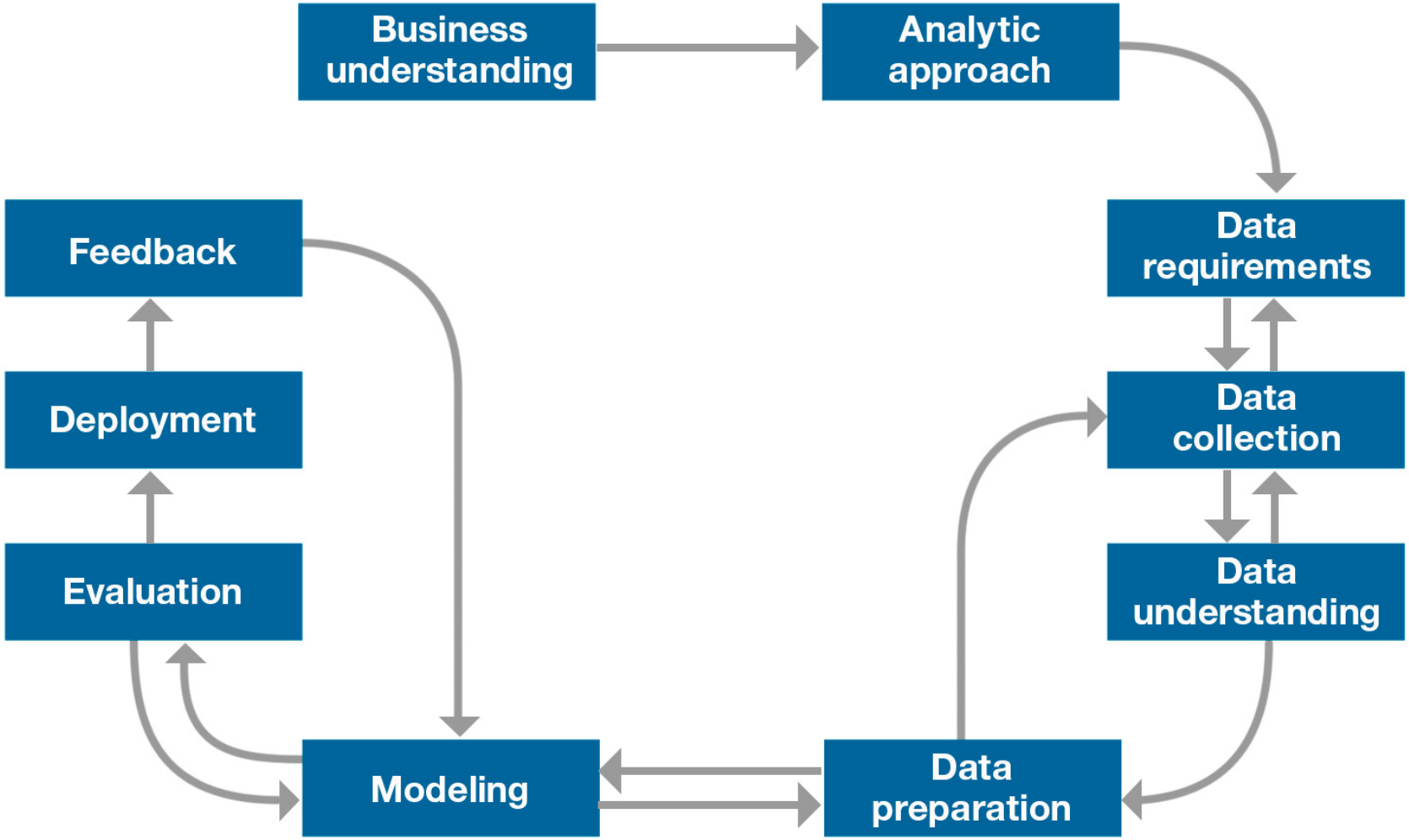
- Ahora ya podemos iniciar el servicio kibana

```
# service kibana start
```

- Lanzamos el navegador
  - http://localhost:5601

Objetivo: Visualización de Datos en Paneles

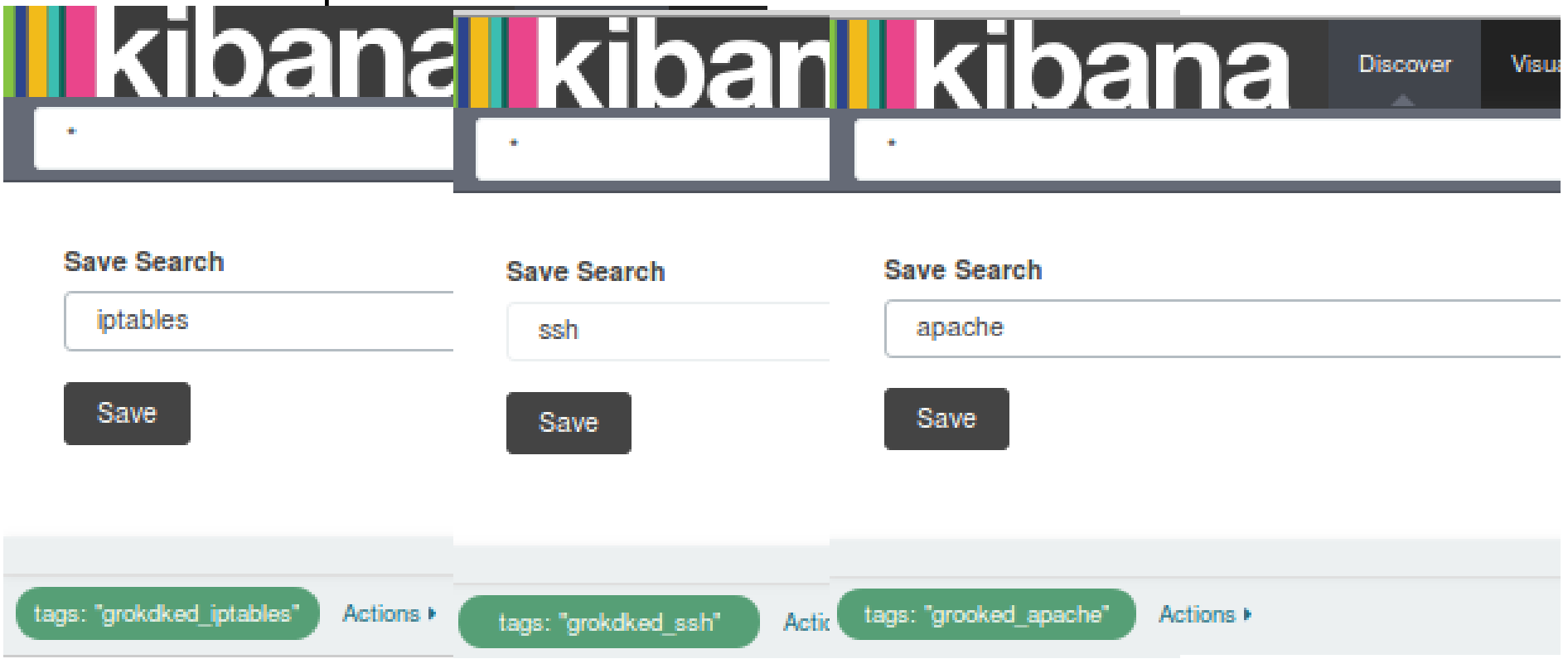
- El orden natural es plantear el problema y determinar si tengo los datos necesarios, para luego normalizarlos y visualizarlos.



Metodología para Data Science
(http://www.ibmbigdatahub.com/blog/why-we-need-methodology-data-science)

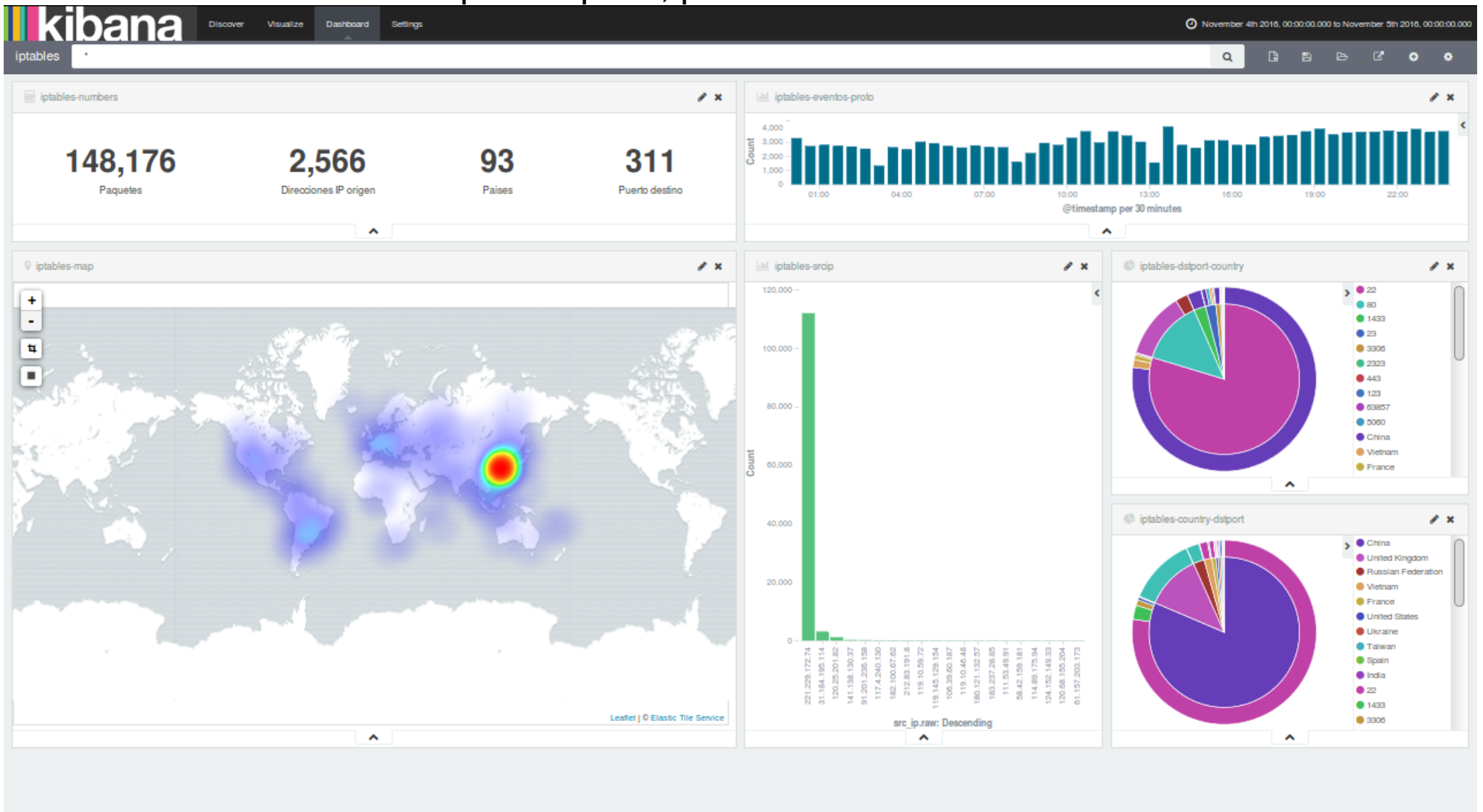
Eligiendo los datos a visualizar: Discover → Searches

- Desde la pestaña “Discover”
  - Crear búsquedas para los datos procedentes de:
    - Iptables
    - SSH
    - Apache

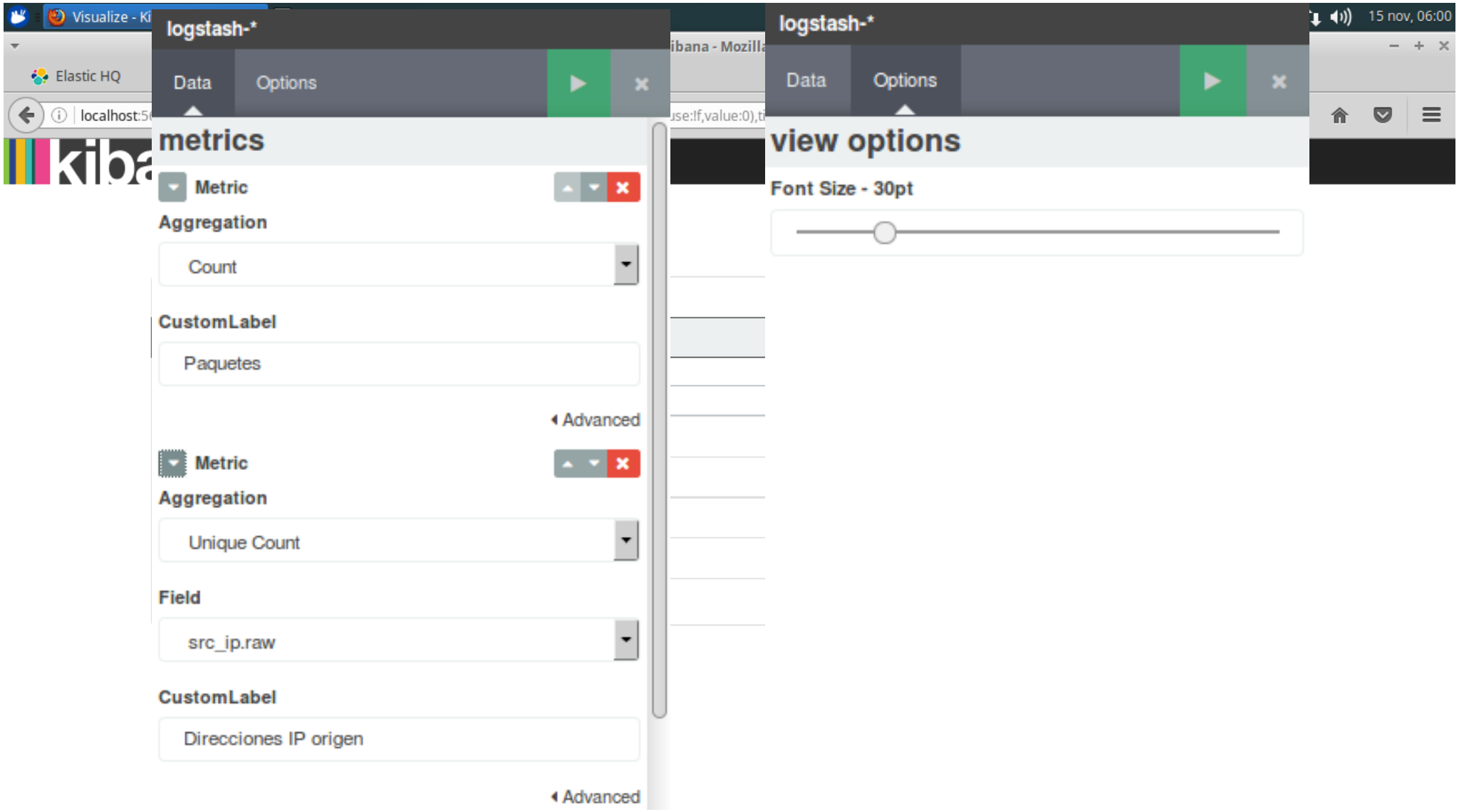


Panel para IpTables

- Objetivo
  - Vamos a crearlo poco a poco, primero las **visualizaciones**



Métricas


logstash-*
Data
Options
metrics
Metric
Aggregation
Count
CustomLabel
Paquetes
Advanced
Metric
Aggregation
Unique Count
Field
src_ip.raw
CustomLabel
Direcciones IP origen
Advanced
logstash-*
Data
Options
view options
Font Size - 30pt

Cuando no es posible contactar con el servidor Elasticsearch

- Basta con comprobar que se está ejecutando el servicio Elasticsearch
  - `# service elasticsearch start`

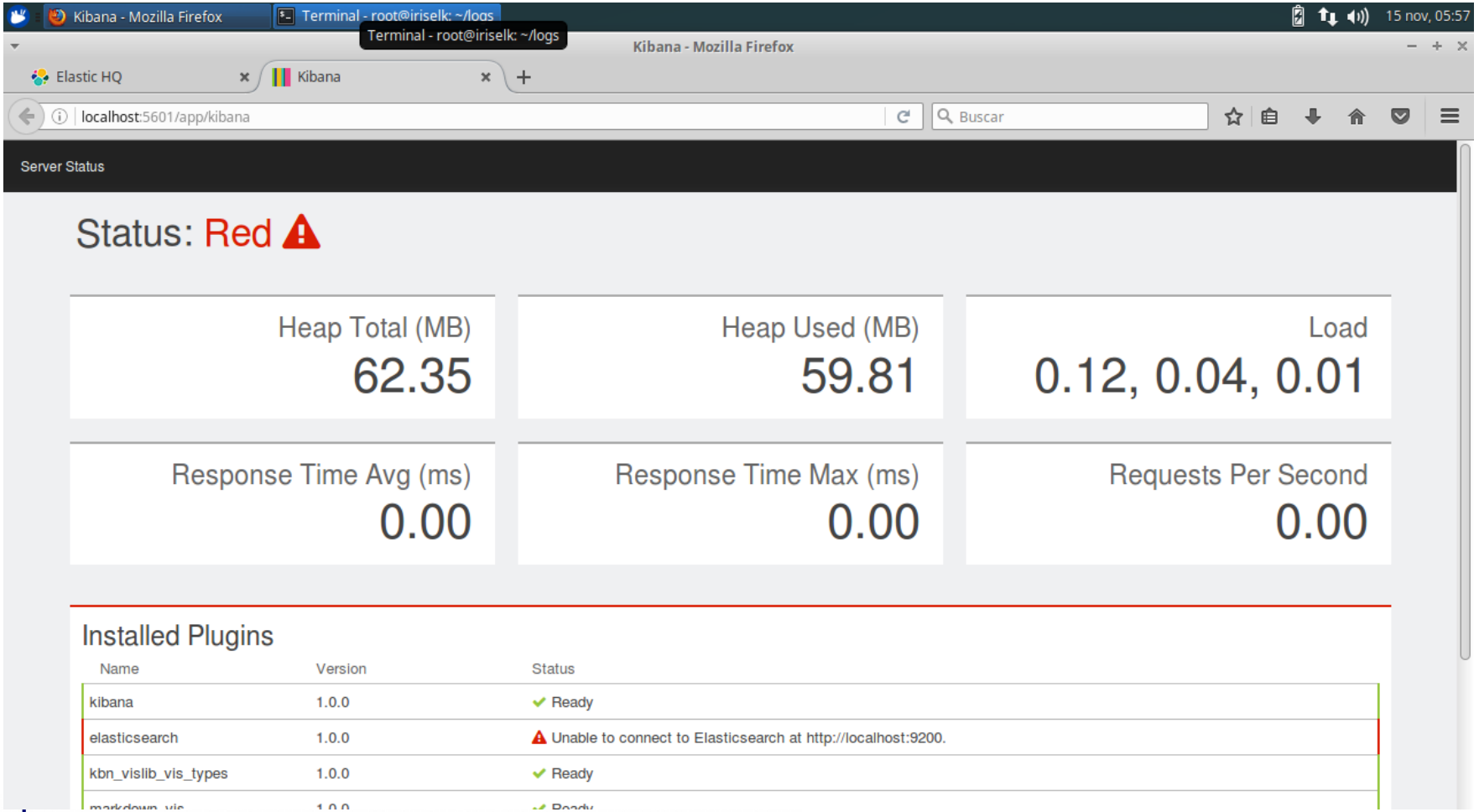
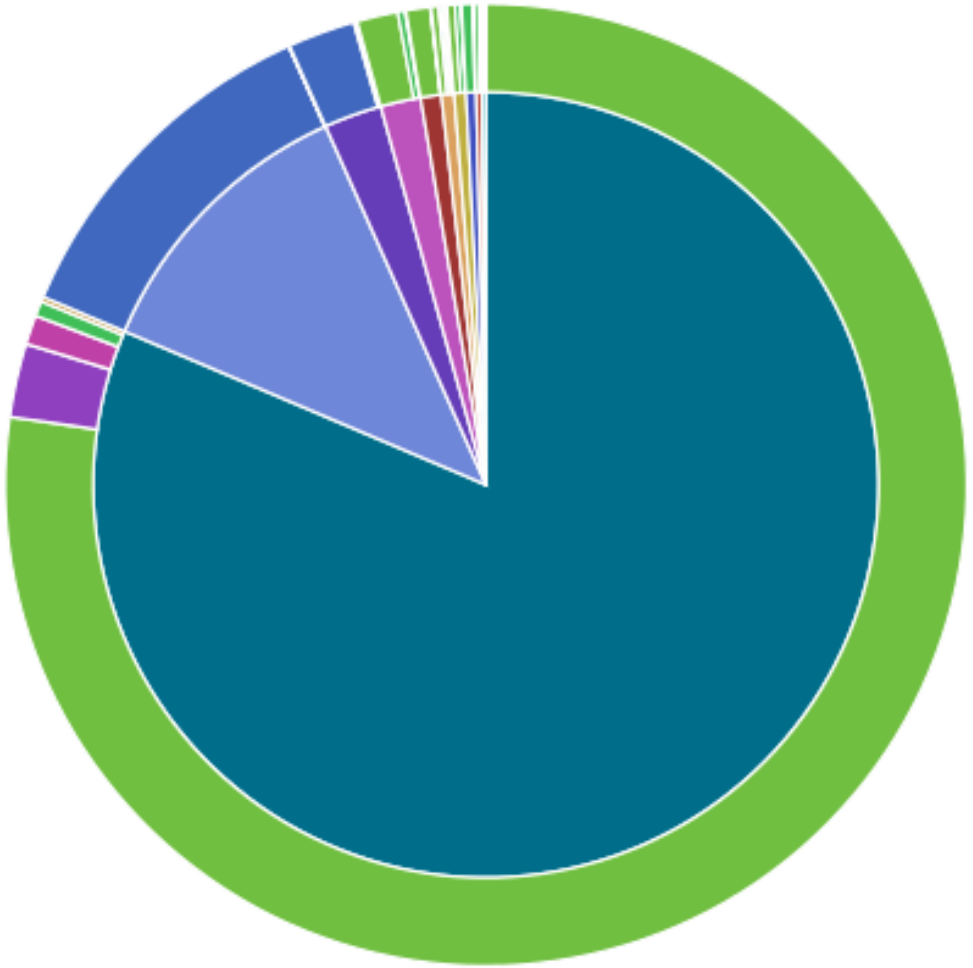
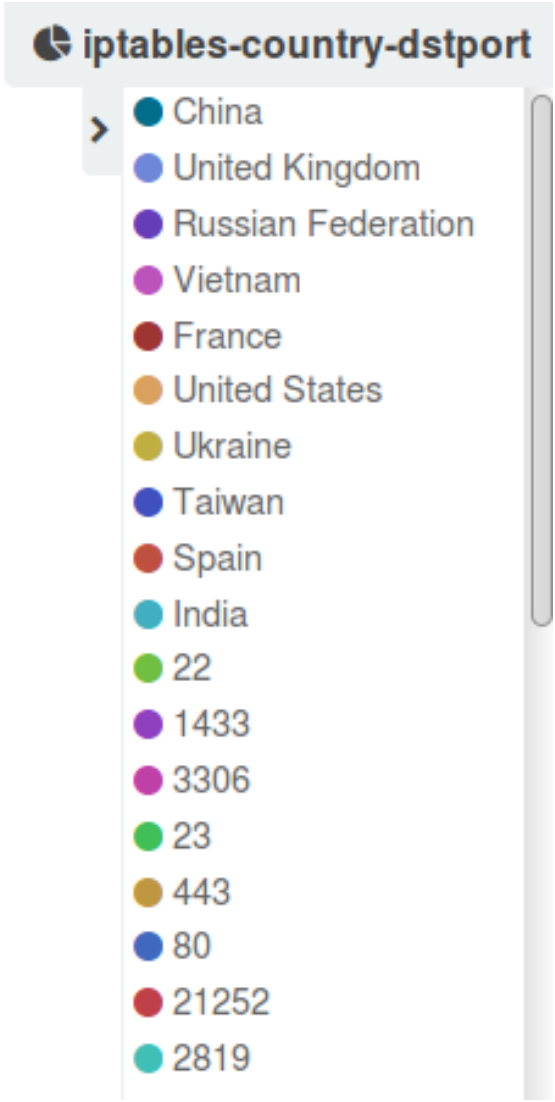
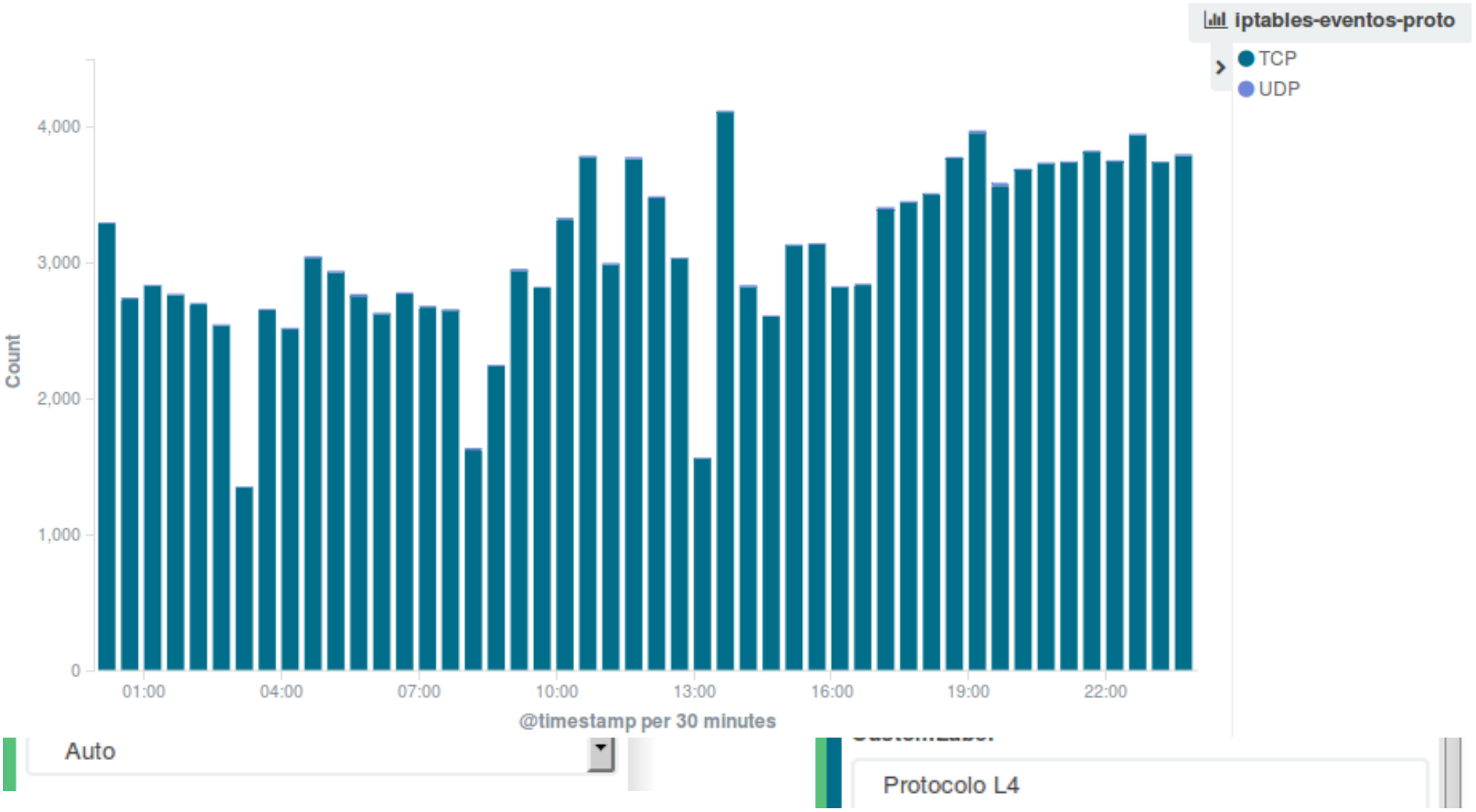


Diagrama de tarta

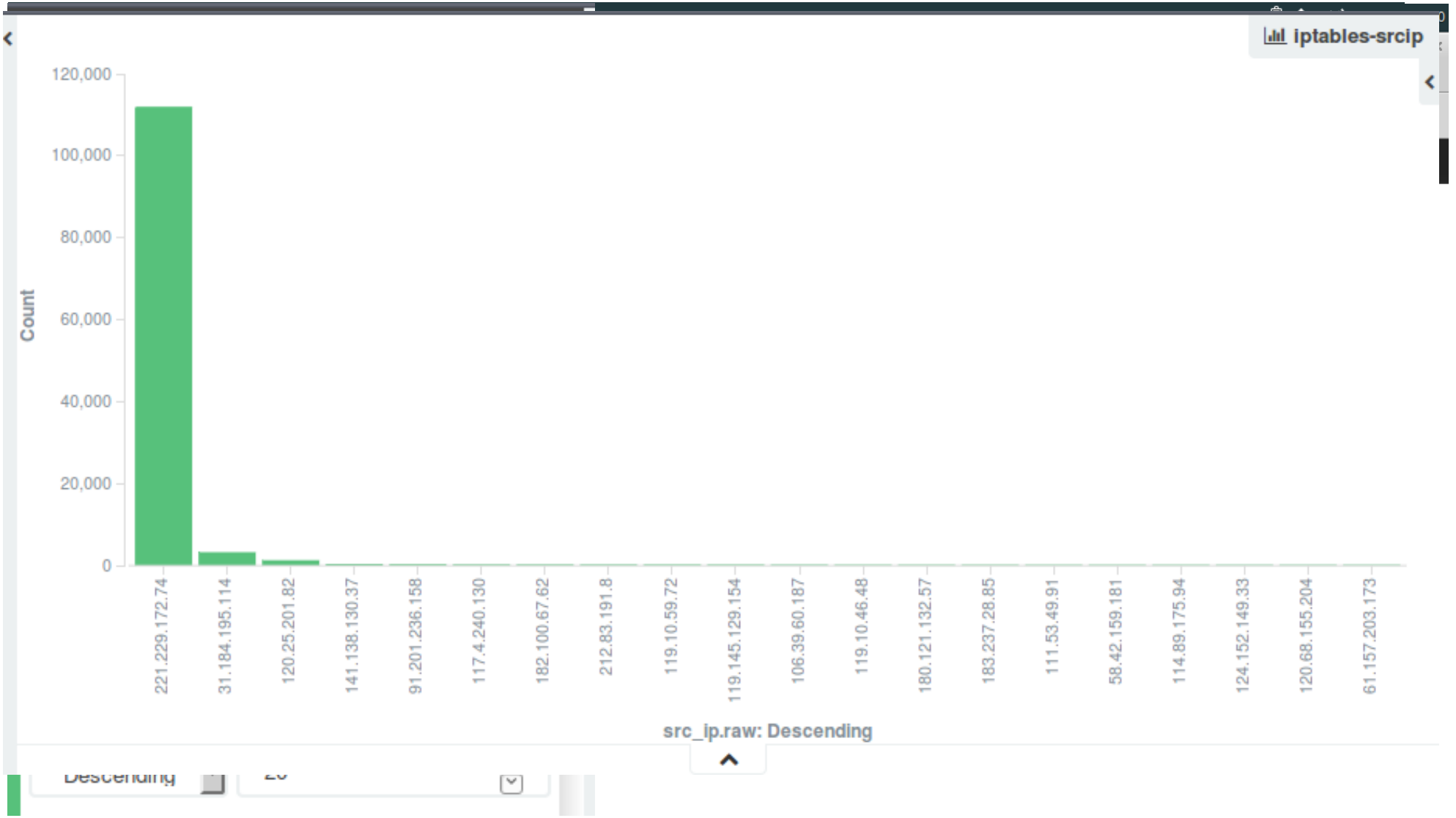




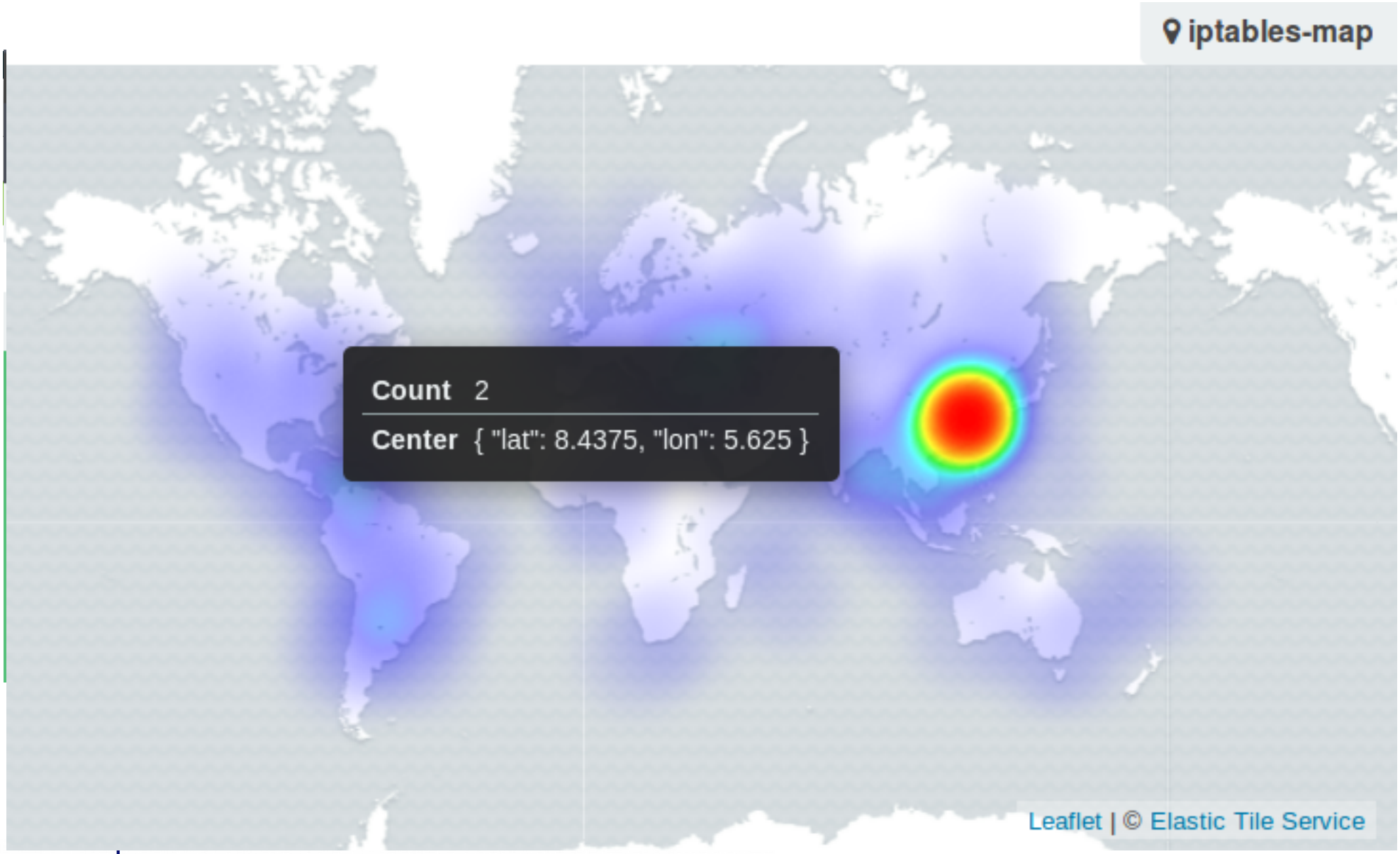
Histogramas Temporales



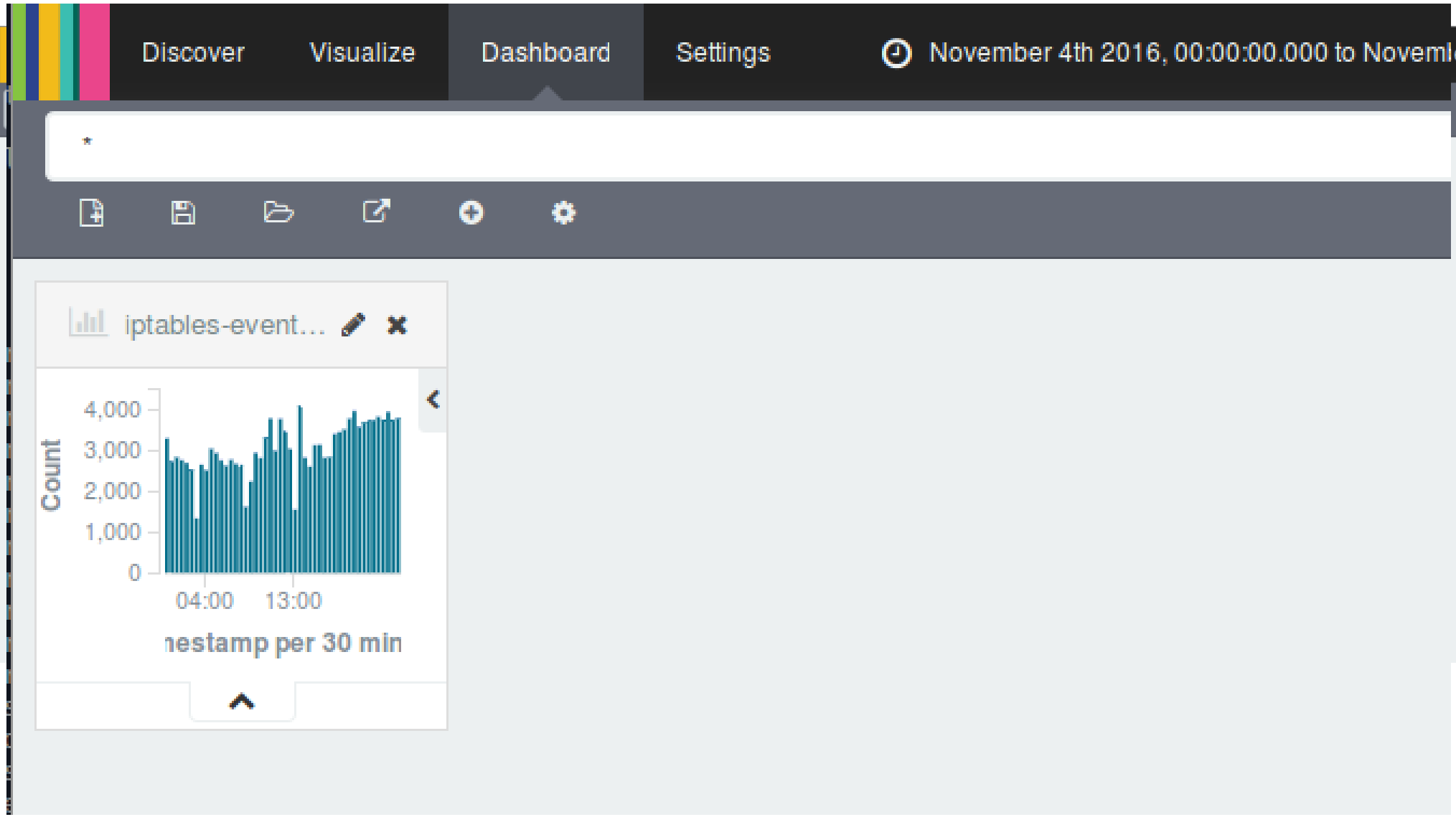
Histogramas sin base temporal



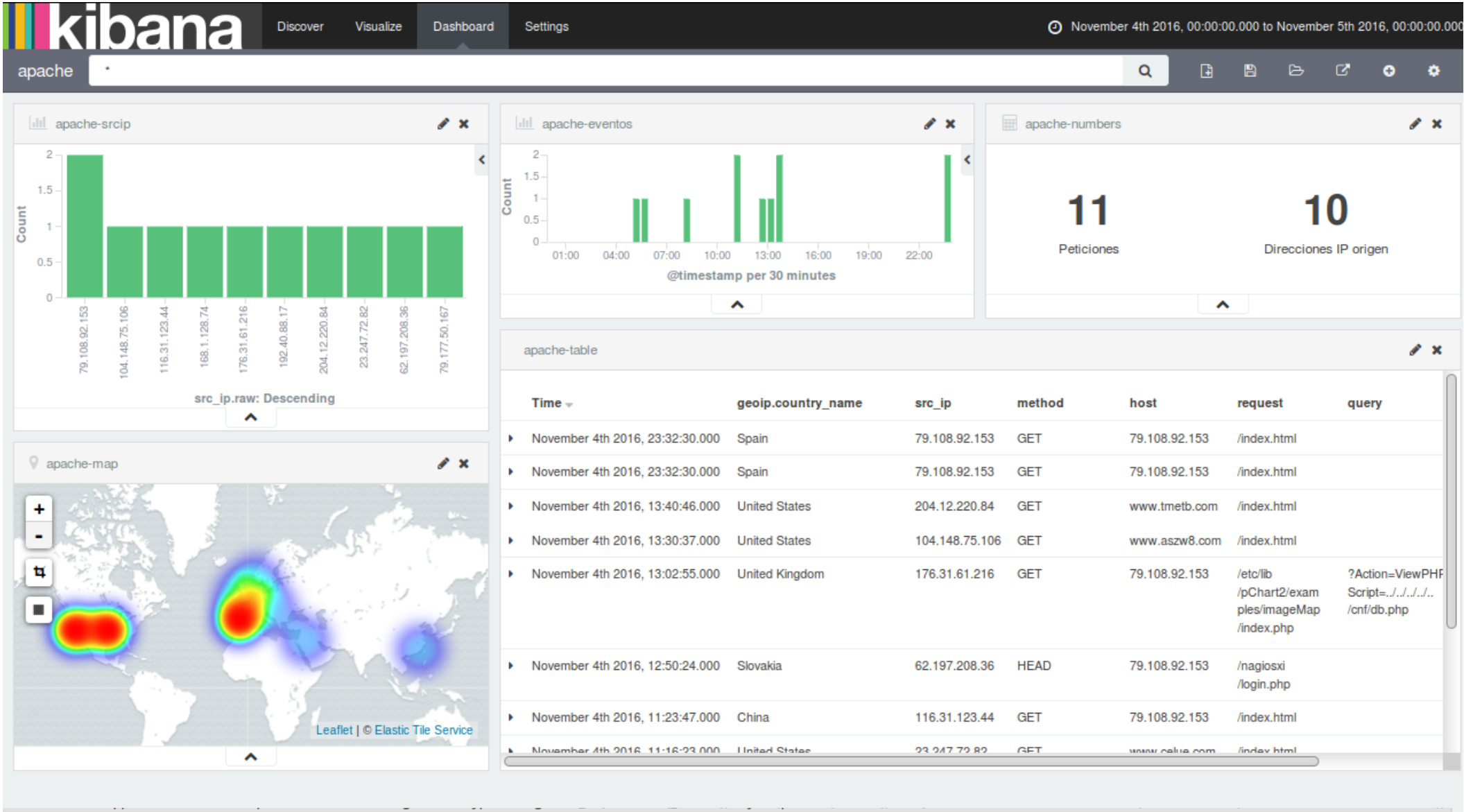
Mapas de calor



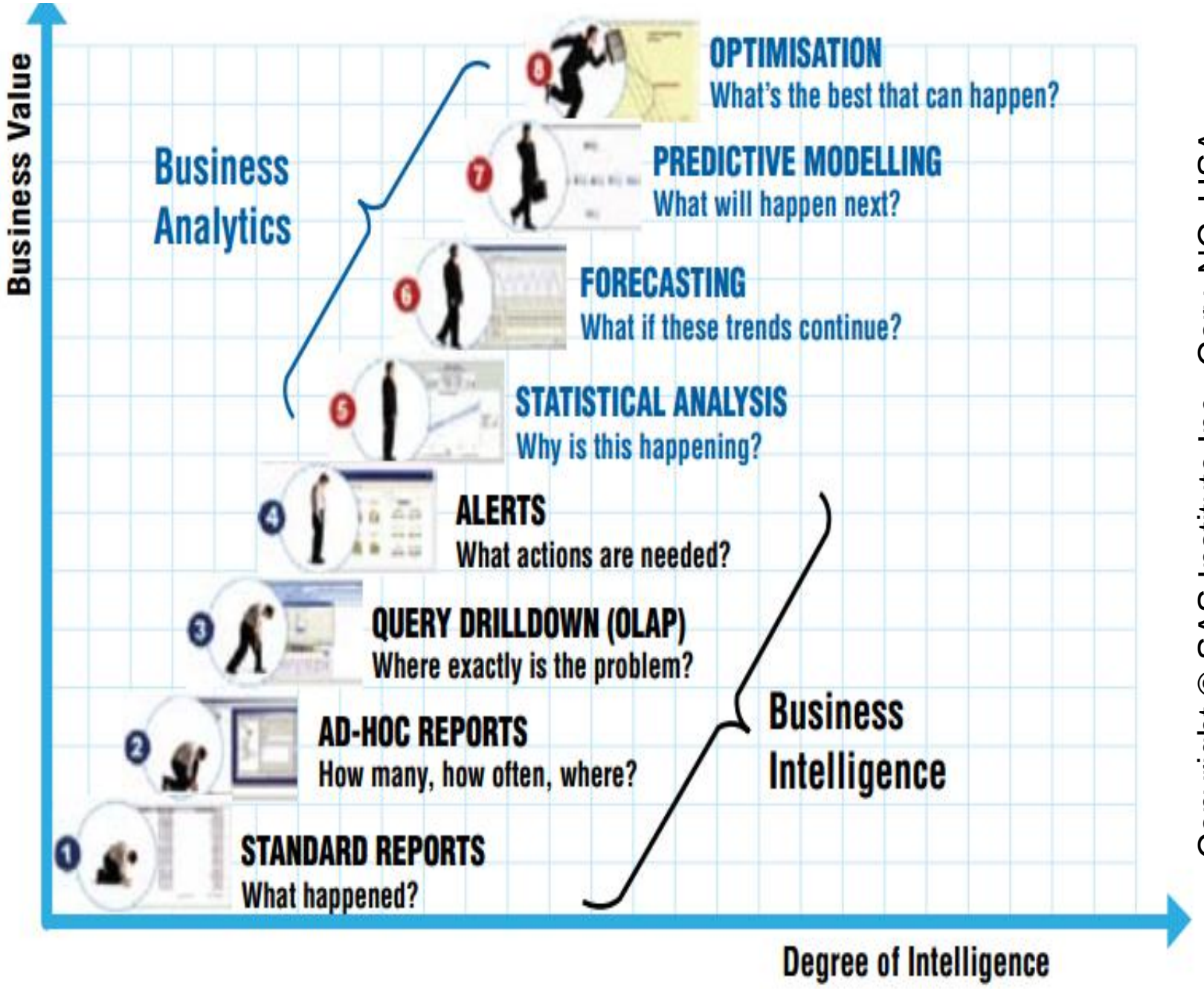
Montamos un panel



Restantes paneles



Modelo de Madurez
¿Dónde estamos?



Conclusiones

- Pasar a “Data driven” (dirigido por datos) no es sencillo
  - Salvo para la Dirección
  - Los datos tienen tantos puntos de vista como usuarios de los mismos
- Se empieza con una maqueta sin presupuesto
  - Y poco a poco se van ganando adeptos
- La demanda crece de forma exponencial
  - Cada usuario aporta sus datos y su visión
  - Los logs de “éxitos” son más importantes que los de fallo
    - Implicaciones en la retención, y almacenamiento necesario
- Objetivo de la Analítica
  - Ya no es la Dirección, como sucede con los Sistemas de Ayuda a la toma de Decisiones (DSS)
  - Se puede beneficiar cualquier persona (alumnos, profesores, personal de administración, etc), además de la Dirección.

- Explotación de logs con ELK
  (Grupo de Usuarios PaloAlto)
  - https://www.youtube.com/watch?v=0fhVNu_egCo
- Gestores de Log
  (Grupos de Trabajo de RedIRIS 2015)
  - http://tv.rediris.es/es/jt2015/492/file/459.html
- Visualización y Analitica de Logs
  (Jornadas Técnicas de RedIRIS 2015)
  - http://tv.rediris.es/es/jt2015/486/file/504.html

Referencias externas

- Elastic WebSite
  - http://www.elastic.co
- Data driven security (libro)
  - http://datadrivensecurity.info/
- “Enhancing Intrusion Analysis through Data Visualization”, Wylie Shanks, 2015. (Artículo)
  - https://www.sans.org/reading-room/whitepapers/detection/enhancing-intrusion-analysis-data-visualization-35757
- Managing events @1M events/s using Elasticsearch (presentación)
  - https://speakerdeck.com/joealex/managing-security-at-1m-events-a-second-using-elasticsearch

Gracias por su atención

Os esperamos en

iris-elk@listserv.uc3m.es

uc3m | Universidad Carlos III de Madrid

Nuria Prieto Pinedo / Rafael Calzada

Grupo de Seguridad

@CertUC3M

cert@uc3m.es