



**Implantación de un sistema de monitorización de red y análisis de logs
en tiempo real para el departamento de informática del IES Fernando
Aguilar Quignon.**

Evaristo R. Rivieccio Vega

1. Estudio del problema y análisis del sistema	5
1.1. Introducción	5
1.2. Objetivos	5
1.3. Funciones y requisitos	5
1.4. Planteamiento y evaluación de diversas soluciones	6
¿Qué es la monitorización?	6
Herramientas de monitorización	8
Nagios	8
Zabbix	10
Pandora FMS	11
Check_MK	12
Splunk	14
ELK Stack (ElasticSearch+Logstash+Kibana+Beats)	15
Elasticsearch	16
Logstash	16
Kibana	17
Beats	17
Grafana	18
Kibana V/s Grafana	18
Prometheus	19
1.5. Justificación y estudio de la solución elegida.	20
Conclusiones sobre las herramientas	20
Extensión X-Pack	21
La batalla elástica	22
Open Distro for Elasticsearch (AWS, Netflix y Expedia)	23
Seguridad	24
Alertas	25
Consulta con SQL	25
Analizador de rendimiento	25
Justificación final de la decisión	26
1.6. Modelado de la solución	28
1.6.1. Recursos utilizados	29
1.7. Planificación temporal	30
2. Arquitectura de los componentes principales	31
2.1 Arquitectura de Elasticsearch	31
Lucene	34

Segmentos de lucene	34
Lucene no borra; actualiza.	36
Tipos de nodos	36
Nodo maestro	36
Nodo de datos	37
Nodo Cliente	37
Nodo coordinador	37
Flujo de consultas de búsqueda en Elasticsearch	37
2.2 Arquitectura de Logstash	38
2.3 Beats - Lumberjack	41
Lumberjack-protocol (El protocolo del leñador)	42
Filebeat - Arquitectura	42
¿Qué es un recolector?	43
¿Qué es un input?	43
3. Fase de pruebas	45
4. Documentación del sistema	46
4.1. Introducción a la aplicación.	46
4.2 Manual de instalación.	46
Elasticsearch	46
RPM (Centos)	46
Ubuntu/Debian	48
Ejemplo de configuración:	50
Kibana	50
RPM (Centos) y Ubuntu/Debian	50
Ejemplo de configuración de Kibana:	51
Beats	52
Planteamiento previo del contexto encontrado	52
Descarga del Beat (Filebeat)	55
Editando la configuración	55
Definiendo Inputs	56
Definiendo Outputs	58
Config para apuntar a Elasticsearch	58
Config para apuntar a Logstash	59
Habilitar un módulo predeterminado (opcional)	59
Instalando paneles en Kibana (Opcional)	60
Arrancando Filebeat	61
Index patterns	61
Ejemplos de paneles predefinidos	65
Paneles Filebeat - módulo system	66
Paneles Filebeat - módulo apache	68
Paneles Metricbeat - módulo system	68

Panes Metricbeat - módulo apache	69
Panes Metricbeat - módulo docker	70
Configuración de ejemplo de metricbeat:	70
APM	70
Logstash	73
Descarga	73
Instalación	74
Configuración	75
Input	75
00inputbeat.conf	76
Filter	77
Dissect	77
Grok	77
10squid.conf	79
System.conf	81
ldap.conf	81
apache.conf	81
89addpersonalizICO.conf	81
Output	81
99out.conf	82
4.3 Manual de administración.	83
Levantar Elasticsearch tras un apagado	83
Errores en Kibana con url demasiado largas	84
Procedimiento de actualización de versión	84
Request Timeout after 300000ms	87
Estado del clúster	87
Error Filebeat exiting	87
PerfTop CLI	87
4.4 Manual de usuario.	90
Discover	90
Búsquedas	93
Kibana Query Language	93
Guardando y abriendo búsquedas	95
Filtrado por campo	97
Administrar filtros	101
Visualizar datos del documento	103
Visualize	104
Dashboards	111
Timelion	113
Alerting	114
Dev Tools	115

Management	115
Index Patterns	116
Saved Objects	117
Advanced Settings	117
Security	117
Tenants	118
Solución a medida del departamento	118
Accesos y navegación de usuarios	119
Panel LDAP	128
Curiosidades	130
Ataque detectado	130
Comparación Docker V/s Máquinas Virtuales	132
QUEMU/KVM:	134
Contenedor Docker	134
Conclusión de la comparación	135
5. Conclusiones finales	135
Anexos	136
Anexo 1	136
Alternativas de código abierto y gratis a X-Pack por componente:	137
Elasticsearch Security	137
Elasticsearch Alerting	137
Elasticsearch Reporting	138
Alternativas de Elasticsearch Graph	138
Elasticsearch Machine Learning	138
Anexo 2	139
Historia completa de ELK Stack	139
6. Bibliografía	143
Acerca de la monitorización	143
Sitios web de herramientas	144
Acerca de Check_MK	144
Acerca de Amazon Elasticsearch	145
Acerca de ELK Stack	145
Acerca de la Arquitectura Elasticsearch:	148
Acerca de Logstash:	148
Acerca de OpenDistro for Elasticsearch:	148
Otros	149

1. Estudio del problema y análisis del sistema

1.1. Introducción

Una herramienta de monitoreo de redes resulta esencial en una corporación o institución, con el objetivo de asegurar el funcionamiento de los sistemas informáticos, así como para evitar cualquier tipo de fallos en la red. Además, este tipo de software resulta de gran ayuda en la optimización de la red ya que aporta un amplio abanico de información sobre el uso de los diferentes recursos de la misma.

Conocer los recursos que están consumiendo los equipos en un momento determinado y poder comparar dicho consumo a lo largo del tiempo puede ser determinante para tomar decisiones en el futuro.

Por lo tanto, se propone implantar una herramienta de monitoreo de redes de la manera más centralizada posible en el departamento de informática del IES Fernando Aguilar Quignon.

Existe un amplio abanico de posibilidades en cuanto a software de monitorización de redes. Se tratará de elegir la opción más práctica, centralizada y cómoda para la posterior administración del sistema de monitoreo.

1.2. Objetivos

Notificación de posibles problemas, ahorro de costes y tiempo así como la optimización de la administración de sistemas mediante:

- Monitorización de servicios
- Monitorización de hosts y dispositivos de red
- Monitorización de logs en tiempo real

1.3. Funciones y requisitos

- Sistema flexible de notificación de eventos.
- Representación gráfica, usabilidad y presentación de los datos en el panel.
- Flexibilidad a la hora de adaptarse a herramientas o software particulares.
- API de acceso desde sistemas externos.
- Auto descubrimiento de servidores y dispositivos de red.
- Agentes nativos en múltiples plataformas.
- Escalado.
- Soporte del mayor número de protocolos de adquisición de datos posible.
- Seguridad física/lógica.

- Integración con máquinas virtuales.
- Integraciones hardware.
- Monitorización de la nube (AWS).
- Gestión de logs centralizada en tiempo real
- Sistema de cruce de datos, permitiendo sacar conclusiones acerca de éstos.

1.4. Planteamiento y evaluación de diversas soluciones

¿Qué es la monitorización?

Monitorización o monitoreo generalmente significa ser consciente del estado de un sistema, para observar una situación de cambios que se pueda producir con el tiempo.

A lo largo de los años “la monitorización” ha sufrido una evolución, por lo que podemos diferenciar sus generaciones:

Evolución y tendencias de las herramientas de monitorización

1.^a Generación – Aplicaciones para monitorizar dispositivos activos o inactivos.

La industria ha desarrollado un sinfín de herramientas para tratar de presentar los recursos de una forma amable y en tiempo real.

Las herramientas de monitorización comenzaron mostrando los elementos a través de un código universal de colores:

-  En verde: todo está funcionando bien.
-  En amarillo: hay algún problema temporal que no afecta la disponibilidad, sin embargo, se deben realizar ajustes para no perder la comunicación.
-  En naranja: el problema se ha hecho persistente y requiere pronta atención para evitar afectaciones a la disponibilidad.
-  En rojo: el dispositivo se encuentra fuera de servicio en este momento y requiere acciones inmediatas para su restablecimiento.

2.^a Generación – Aplicaciones de análisis de métricas

En esta generación las herramientas realizan un análisis a profundidad con el fin de poder evaluar los estados de los componentes dentro de los dispositivos (CPU, memoria, espacio de almacenamiento, paquetes enviados y recibidos, broadcast, multicast, etc.) De tal manera que permita ajustar los parámetros y evaluar los niveles de servicio del dispositivo.

Este tipo de aplicaciones se apoyan en analizadores de protocolos o “sniffers” y en elementos físicos distribuidos conocidos como “probes” (sondas) para recolectar estadísticas del tráfico, que son controlados típicamente desde una consola central.

3.^a Generación – Aplicaciones de análisis optimizado

Esta generación de aplicaciones con enfoque transaccional captura ahora “flujos” de tráfico e identifica cuellos de botella y latencias a lo largo de las conexiones que existen entre los componentes de un servicio, entregando información acerca de la salud del mismo.

Con esta generación se logra conectar todas las partes de manera eficiente, donde cada dispositivo sabe cuándo se debe informar a otro dispositivos sin afectar en las tareas que esté realizando, evitando así una sobrecarga de información.

4.^a. Generación – Indicadores gráficos - “Dashboards”

Llevando el crecimiento de las soluciones tecnológicas a los requerimientos de las organizaciones de hoy, llegamos a las vistas de “dashboard” que son indicadores que el cliente puede crear y personalizar de acuerdo a sus necesidades, además de poder seleccionar las variables necesarias para correlacionar datos que ayuden a la toma de decisiones.

Según el objetivo de la monitorización podemos diferenciar dos tipos de monitorización: la monitorización predictiva, que nos permite tomar decisiones; y la monitorización proactiva, que es la que se encarga de detectar y solucionar los problemas encontrados.

- **Monitorización predictiva**

- Ayuda a anticiparnos al problema.
- Ofrece datos reales de la plataforma.
- Permite tomar decisiones.
- Posibilita un trabajo de revisión continuada.

- **Monitorización proactiva**

- Detecta los problemas.
- Soluciona los problemas.
- Dar respuestas al día a día.
- Suplir los problemas por falta de personal.

Herramientas de monitorización

Se han seleccionado las herramientas que, según un estudio previo, podrían satisfacer mejor las necesidades expuestas en los puntos anteriores. Primero se describirán las herramientas que descartamos por las razones expuestas en cada caso:

Nagios®

Nagios

Nagios¹, llamado originalmente Netsaint, nombre que se debió cambiar por coincidencia con otra marca comercial, fue creado y es actualmente mantenido por Ethan Galstad, junto con un grupo de desarrolladores de software que mantienen también varios complementos.

nagios	
General	
• Home	
• Documentation	
Monitoring	
• Tactical Overview	
• Service Detail	
• Host Detail	
• Status Overview	
• Status Summary	
• Status Grid	
• Status Map	
• S-D Status Map	
• Service Problems	
• Host Problems	
• Network Outages	
• Comments	
• Downtime	
• Process Info	
• Performance Info	
• Scheduling Queue	
Reporting	
• Trends	
• Availability	
• Alert Histogram	
• Alert History	
• Alert Summary	
• Notifications	
• Event Log	
Configuration	
• View Config	

webprod03	Critical	01-26-2007 14:59:59	0d 0h 5m 23s	1/4	
Check_Users	OK	01-26-2007 14:59:54	0d 0h 5m 23s	1/4	USERS OK - 1 users currently logged in
Current_Load	OK	01-26-2007 14:59:54	0d 0h 5m 23s	1/4	OK - load average: 0.21, 0.00, 0.05
Memory_Usage	OK	01-26-2007 14:59:29	0d 0h 5m 23s	1/4	OK: Memory Usage 56% - Total: 511 MB, Used: 287 MB, Free: 224 MB
PING	OK	01-26-2007 14:59:14	0d 0h 5m 23s	1/4	PING OK - Packet loss = 0%, RTA = 0.0 ms
Root_Partition	OK	01-26-2007 14:59:09	0d 0h 5m 23s	1/4	DISK OK ([238916 kb (0%) free on /dev/sda2])
SWAP_Usage	OK	01-26-2007 14:57:44	0d 0h 5m 23s	1/4	Swap ok - (null) 0% (0 out of 16384)
Total_Processes	OK	01-26-2007 14:59:29	0d 0h 5m 23s	1/4	OK - 293 processes running
Xen_Virtual_Machine_Monitor	Critical	01-26-2007 14:59:04	0d 0h 4m 34s	1/4	Critical Xen VMs Usage - Total NB: 0 - detected VMs: migrating-xen-vms
webprod04	OK	01-26-2007 14:59:54	0d 0h 15m 33s	1/4	
Check_Users	OK	01-26-2007 14:59:54	0d 0h 15m 33s	1/4	USERS OK - 2 users currently logged in
Current_Load	OK	01-26-2007 14:59:34	0d 0h 15m 33s	1/4	OK - load average: 0.31, 0.00, 0.44
Memory_Usage	OK	01-26-2007 14:59:19	0d 0h 14m 13s	1/4	OK: Memory Usage 37% - Total: 511 MB, Used: 190 MB, Free: 321 MB
PING	OK	01-26-2007 14:59:19	0d 0h 14m 13s	1/4	PING OK - Packet loss = 0%, RTA = 0.27 ms
Root_Partition	OK	01-26-2007 14:57:08	0d 0h 14m 13s	1/4	DISK OK ([4189880 kb (0%) free on /dev/sda2])
SWAP_Usage	OK	01-26-2007 14:58:34	0d 0h 15m 33s	1/4	Swap ok - (null) 0% (0 out of 16384)
Total_Processes	OK	01-26-2007 14:59:09	0d 0h 15m 22s	1/4	OK - 293 processes running
Xen_Virtual_Machine_Monitor	Warning	01-26-2007 14:58:54	0d 0h 1m 33s	1/4	Warning Xen VMs Usage - Total NB: 1 - detected VMs: migrating-xen-vms
webprod05	OK	01-26-2007 14:59:39	0d 0h 24m 05s	1/4	
PING	OK	01-26-2007 14:59:34	0d 0h 24m 05s	1/4	PING OK - Packet loss = 0%, RTA = 0.25 ms
Xen_Virtual_Machine_Monitor	OK	01-26-2007 14:59:34	0d 0h 24m 05s	1/4	Xen Hypervisor "webprod05" is running 4 Xen VMs: xen-vm1, xen-vm2, xen-vm3, xen-vm4
xen-vm1	OK	01-26-2007 14:58:09	0d 0h 17m 29s	1/4	
Check_Users	OK	01-26-2007 14:58:09	0d 0h 17m 29s	1/4	USERS OK - 1 users currently logged in
Current_Load	OK	01-26-2007 14:57:54	0d 0h 16m 21s	1/4	OK - load average: 1.51, 0.05, 0.49
Memory_Usage	OK	01-26-2007 14:59:39	0d 0h 15m 41s	1/4	OK: Memory Usage 8% - Total: 8198 MB, Used: 676 MB, Free: 7519 MB
PING	OK	01-26-2007 14:59:39	0d 0h 15m 41s	1/4	PING OK - Packet loss = 0%, RTA = 0.49 ms
Root_Partition	OK	01-26-2007 14:59:59	0d 0h 15m 41s	1/4	DISK OK ([4189880 kb (0%) free on user])
SWAP_Usage	OK	01-26-2007 14:59:44	0d 0h 15m 41s	1/4	Swap ok - (null) 0% (0 out of 2005)
Total_Processes	OK	01-26-2007 14:57:29	0d 0h 18m 3s	1/4	OK - 88 processes running
xen-vm2	OK	01-26-2007 14:57:15	0d 0h 7m 41s	1/4	
Check_Users	OK	01-26-2007 14:57:15	0d 0h 7m 41s	1/4	USERS OK - 0 users currently logged in
Current_Load	OK	01-26-2007 14:57:59	0d 0h 7m 1s	1/4	OK - load average: 0.00, 0.00, 0.00
Memory_Usage	OK	01-26-2007 14:59:44	0d 0h 6m 21s	1/4	OK: Memory Usage 6% - Total: 1023 MB, Used: 64 MB, Free: 959 MB
PING	OK	01-26-2007 14:59:19	0d 0h 6m 14s	1/4	PING OK - Packet loss = 0%, RTA = 0.43 ms
Root_Partition	OK	01-26-2007 15:00:05	0d 0h 15m 4s	1/4	DISK OK ([524220 kb (99%) free on user])
SWAP_Usage	OK	01-26-2007 14:55:49	0d 0h 6m 41s	1/4	Swap ok - (null) 0% (0 out of 2055)
Total_Processes	OK	01-26-2007 14:56:34	0d 0h 5m 1s	1/4	OK - 52 processes running

los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado.

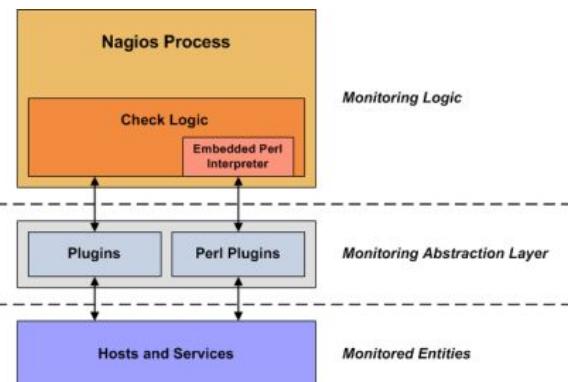
Está escrito en C y entre sus características principales figuran la monitorización de servicios de red (SMTP, POP3, HTTP, SNMP...), la monitorización de los recursos de sistemas hardware (carga del procesador, uso de los discos, memoria, estado de los puertos...), independencia de sistemas operativos, posibilidad de monitorización remota mediante túneles SSL cifrados o SSH, y la posibilidad de programar plugins específicos para nuevos sistemas.

Nagios, licenciado bajo la GNU General Public License Version 2, es un sistema de monitorización de redes ampliamente utilizado, que vigila

¹ Sítio web de Nagios: <https://www.nagios.com/>

Nagios permite resolver problemas de forma rápida mediante manejo de eventos, de forma que ante un evento detectado puede desencadenar acciones, como por ejemplo, levantar otro servidor si el estado de la máquina es crítico.

Ha sido el estándar de la industria de la monitorización desde 1999, pero actualmente existen opciones muchísimo mejores.



Las motivaciones que nos han llevado a descartarlo son:

- Nagios base está extremadamente limitado en funcionalidades de serie, teniendo que suplir esta carencia con plugins, addons y extensiones de terceros.
- Nagios no está pensado para entornos cambiantes, su configuración es estática, engorrosa y complicada de integrar en procesos de provisión automática. Como es bien sabido, la escalabilidad no es el punto fuerte de Nagios.



Zabbix

(GNU GPL)

Zabbix² es un Sistema de Monitorización de Redes creado por Alexei Vladishev. Está diseñado para monitorizar y registrar el estado de servicios de red, Servidores, y hardware de red.

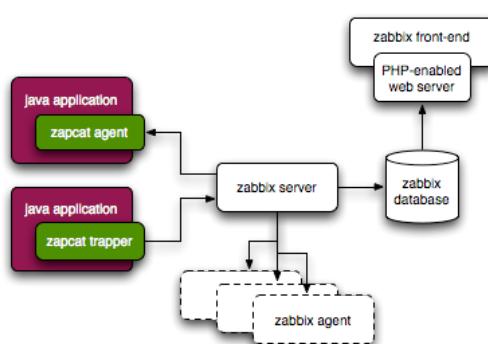
Se comenta que en muchos casos hace la sombra a Nagios.

Zabbix surge en 2001. Es un desarrollo completo, no un fork de Nagios, y su principal característica es que tiene una visión más holística de la monitorización, cubriendo rendimiento, no solo estados, ya que esta es una de las carencias más significativas de Nagios. Además de disponer de un sistema de gestión WEB que permite gestionarlo de forma centralizada, sin farragosos ficheros de configuración, como le pasaba a Nagios.

El núcleo de Zabbix está escrito en C, mientras que la interfaz gráfica de usuario web está escrita en PHP. Almacena los datos en una base de datos relacional.

Al igual que Nagios, lo descartamos, entre otros motivos, por los siguientes:

- Hay constancia de que Zabbix tiene una degradación del rendimiento a partir de 1,000 nodos, debido a las limitaciones del front-end de PHP y la GUI web.
- Zabbix no tiene informes en tiempo real; y sólo es compatible con una variedad de mecanismos de notificación, en tiempo casi real. El monitoreo de



² Sítio web de Zabbix: <https://www.zabbix.com/>

ficheros de logs tampoco es tan bueno. Por otra parte, la configuración, aunque intuitiva, requiere de muchos clics y pasos para completarla.

- Aunque a nivel técnico es muy potente, Zabbix requiere de la instalación de numerosos plugins para ser eficiente y poder alcanzar sus funcionalidades completas.



Pandora FMS

Pandora FMS³ es un software de código abierto que sirve para monitorizar y medir todo tipo de elementos.

Pandora FMS tiene su origen en el año 2004. También surge como un desarrollo nuevo que parte desde cero, buscando cubrir todos los aspectos posibles de la monitorización, desde infraestructura (redes y servidores), hasta experiencia web, negocio, rendimiento y aplicaciones.



El núcleo de Pandora está escrito en Perl, mientras que la interfaz gráfica de usuario web está escrita en PHP y javascript. Almacena los datos en una base de datos relacional.

Parece una herramienta muy potente, más optimizada que Zabbix, con una apariencia más agradable.

Pero indagando más nos damos cuenta de que ésto es sólo en su versión Enterprise, ya que la versión de la comunidad está muy limitada⁴. Entre otras limitaciones, encontramos: la incapacidad de recolectar logs, la ausencia de autenticación LDAP, etc... motivo por el cual descartamos también ésta opción.

³ Sítio web de pandora: <https://pandorafms.com/es/>

⁴ Limitaciones de V. Community <https://pandorafms.com/es/precios-de-pandora-fms/#pricing%7C1>

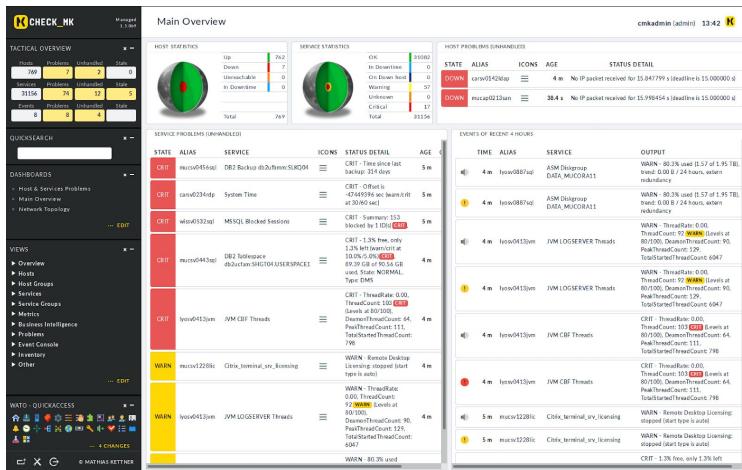


CHECK_MK

Das Monitoring

Check_MK

Check_MK⁵ fue en sus primeros años de vida un plugin de Nagios, que aportaba muchas mejoras en funcionalidad y rendimiento a la herramienta. Llegó a tener tal popularidad, que pasó a ser una herramienta independiente. En abril de 2009 se lanzó bajo la GPL.

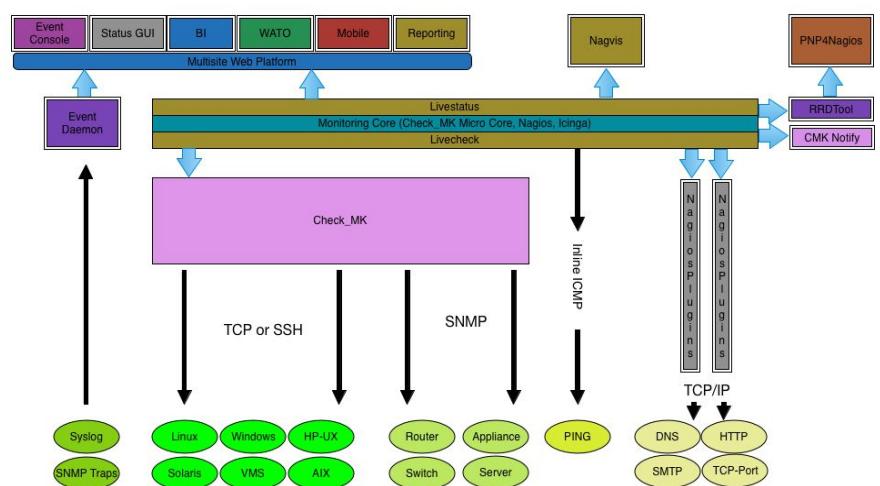


The screenshot shows the main monitoring interface. On the left, there's a sidebar with navigation links like 'Main Overview', 'Service Problems (Unhandled)', 'Event Log (Recent 4 Hours)', and 'Views'. The main area has three circular status indicators: Host Statistics (769 hosts), Service Statistics (769 services), and Host Problems (769). Below these are sections for 'SERVICE PROBLEMS (UNHANDLED)' and 'EVENTS IN RECENT 4 HOURS'. The 'SERVICE PROBLEMS' section lists several critical (CRIT) and warning (WARN) events. The 'EVENTS IN RECENT 4 HOURS' section shows a log of events from various services, each with a timestamp, alias, service name, output, and severity level.

Utiliza una forma diferente para configurar los objetos respecto a Nagios, siendo más potente y funcional que el estándar. Para aprovecharlo es necesario instalar los agentes de monitorización en los elementos a monitorizar. De esta forma, nos permite añadir una serie de chequeos de forma automática.

Permite conocer el estado de los elementos monitorizados en tiempo real de forma rápida y funcional.

Hasta ahora la forma de acceder al estado de los objetos de Nagios era leyendo un fichero o consultando una base de datos. Check_MK crea un



⁵ Sítio web de Check_MK: <https://checkmk.de/>

socket que permite sin apenas consumir recursos, saber de forma rápida el estado de nuestra infraestructura.

Ofrece una versión Enterprise con varias mejoras y comodidades, como exportación de gráficos, y un núcleo más optimizado aún, pero la versión gratuita es 100% funcional.

Habíamos comentado anteriormente como Zabbix suponía una mejora contra Nagios, nos preguntamos entonces: ¿Supone Check_MK una mejora contra Zabbix?

La respuesta es sí, sobre todo en los siguientes puntos de Check_MK:

- Configuración de ficheros planos.
 - Facilidad para implementar scripts⁶ y/o agentes⁷ personalizados escritos en Python, en los cuales dependiendo de la salida del script (0,1,2) manejaremos los: OK, CRITICAL o WARNING.
- Mejor inventario automatizado.
- Detección de “Flapping”⁸ suspendiendo inmediatamente las alertas para no saturar con “ruido” la bandeja de entrada de incidencias.
- Agentes muy ligeros.
- Gráficos más rápidos.
- Madurez de proyecto.
- Documentación⁹ de calidad; muy detallada y completa.

Una herramienta perfecta para monitorizar servicios y hosts. En cuanto a la funcionalidad que también deseamos, de monitorizar ficheros de log, encontramos que tenemos un plugin para Check_MK llamado “mk_logwatch¹⁰” gracias al cual podríamos rastrear ciertas cadenas clave como “ERROR” por ejemplo, en los ficheros en los que nos interesa.

Una vez configurado, cada vez que apareciera una línea con dicha palabra clave, en el fichero de log que está siendo monitorizado, nos aparecería un WARNING, o CRITICAL según hayamos decidido.

Nos valdría para ser alertados de ciertos eventos que consideremos de interés, pero se nos queda corto, echaríamos de menos la posibilidad de poder tratar de manera

⁶ Plugin para habilitar Telegram como método de alerta en Check_MK implementado durante la FCT:
<https://suevaristo.blogspot.com/2019/04/bot-telegram-para-recibir.html>

⁷ Script/agente customizado para monitorizar el estado de las alertas de Amazon CloudWatch implementado durante la FCT:
<https://suevaristo.blogspot.com/2019/05/monitoreando-la-monitorizacion.html>

⁸ Flapping: Estado de una alerta en el que la misma comienza a cambiar entre dos o más estados rápidamente, durante un corto periodo de tiempo.

⁹ Documentación oficial de Check_MK: <https://checkmk.com/cms.html>

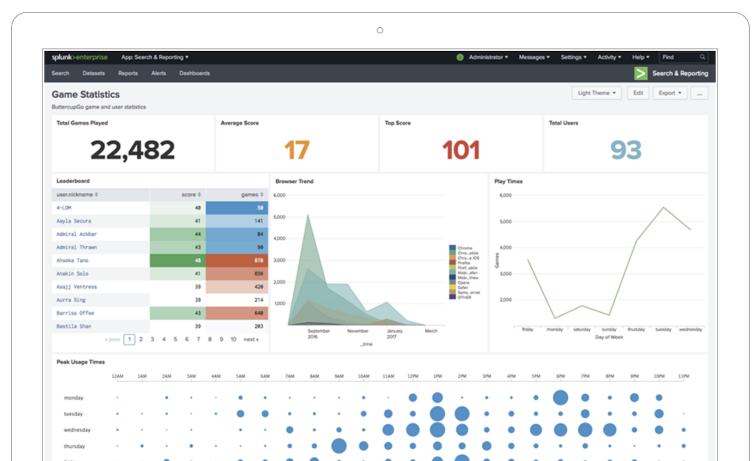
¹⁰ mk_logwatch: https://checkmk.de/checkmk_logfiles.html

centralizada los logs de todos los equipos del departamento, pudiendo “jugar” con los datos, para sacar conclusiones; y éste no es el objetivo de Check_MK, no fue diseñado para ello, por lo que, pese a ser la mejor herramienta de monitorización de las que hemos comentado, vamos a ir más allá, centrándonos en herramientas que sí estén diseñadas en su origen para tratar con ficheros de logs, de forma avanzada.



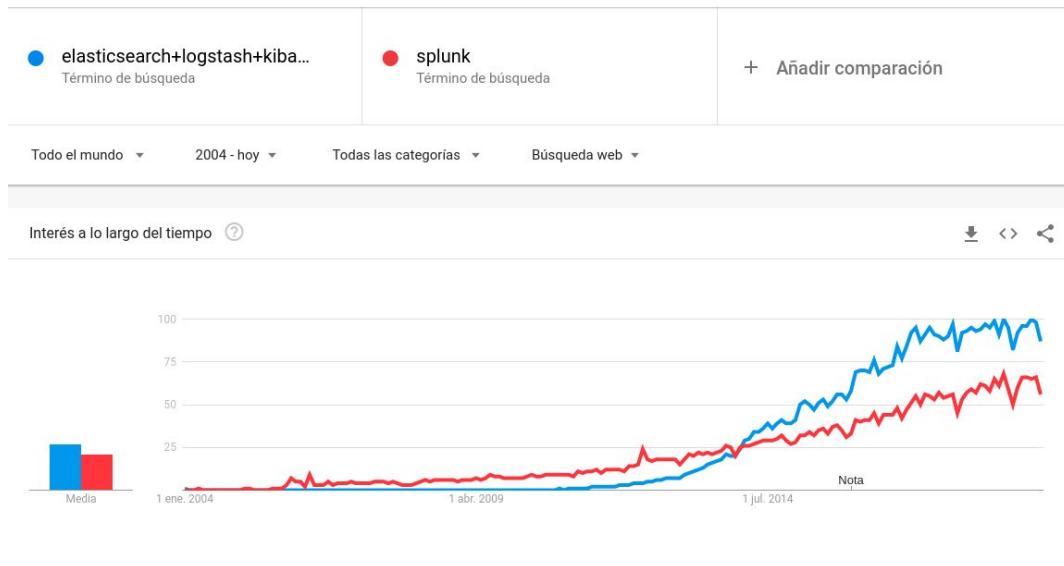
Splunk

Splunk¹¹ fue una de las primeras compañías en lidiar con los problemas inherentes al registro y los datos de la máquina, incluso antes de que se acuñara el término big data. Fundada en 2003 (Michael Baum, Rob Das y Erik Swan), el origen de su nombre proviene de "espeleología", que es la práctica de explorar cuevas. No es de código abierto, pero nuestra investigación parte de conocer la existencia de Splunk, aunque no entremos en detalles de su arquitectura. “La otra cara de la moneda” es ElasticSearch, que sí era de código abierto, y que fue lanzada por Shay Banon en 2010. En base a Elasticsearch, se fundó una compañía, ahora llamada Elastic, que ofrece un conjunto de herramientas, en un producto llamado ELK Stack. Estas herramientas son Elasticsearch, Logstash, Kibana, y de recién incorporación a la familia, Beats.

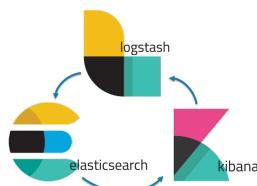


¹¹ Sitio web de Splunk: <https://www.splunk.com>

Podemos observar en las tendencias de búsqueda de Google, como elasticsearch+logstash+kibana ha terminado superando a Splunk en búsquedas en los últimos años.



Es por ésto, que a continuación analizaremos ELK Stack.



ELK Stack (ElasticSearch+Logstash+Kibana+Beats)

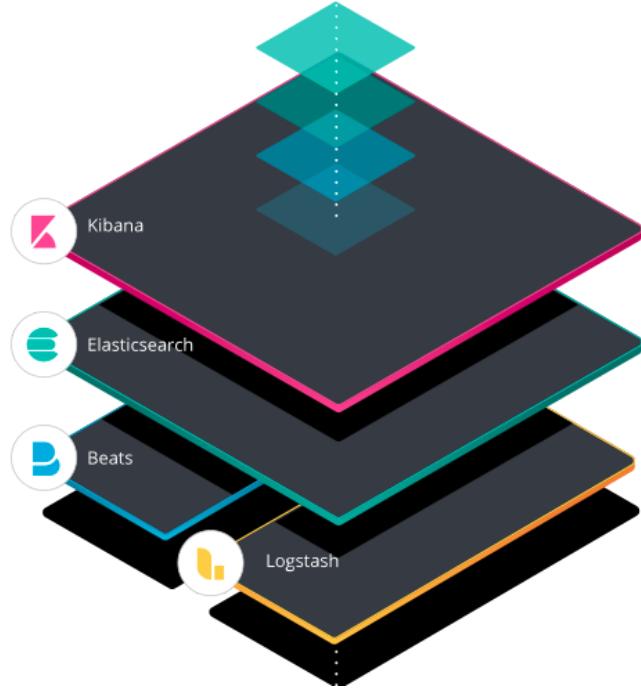
ELK Stack¹² es un conjunto de herramientas de gran potencial de código abierto (Licencia Apache) que se combinan para crear una herramienta de administración de registros permitiendo la monitorización, consolidación y análisis de logs generados en múltiples servidores.

¹² Sitio Web de ELK Stack: <https://www.elastic.co/elk-stack>



Elasticsearch

Elasticsearch es un motor de búsqueda y análisis RESTful distribuido, capaz de resolver un número creciente de casos de uso.



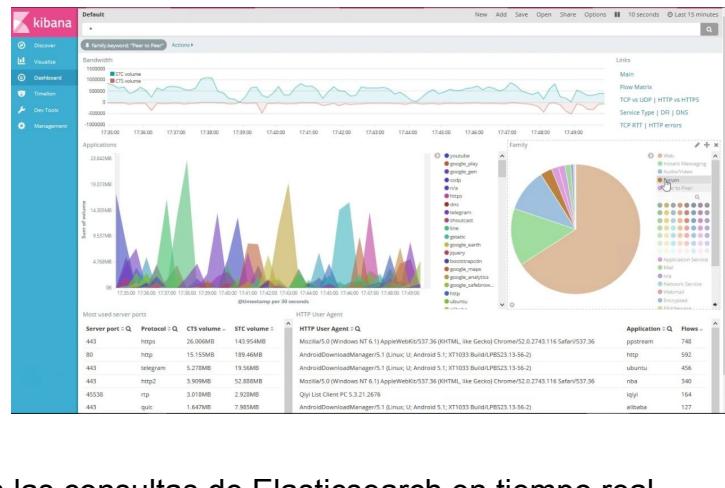
Elasticsearch permite realizar y combinar muchos tipos de búsquedas: estructuradas, no estructuradas, geográficas, métricas, etc... almacenando los datos de forma descentralizada. Está basado en Java, lo que permite que Elasticsearch se ejecute en diferentes plataformas. Permite a los usuarios explorar una gran cantidad de datos a muy alta velocidad.



Logstash es una fuente de procesamiento de datos del lado del servidor de código abierto que ingiere datos de una multitud de fuentes simultáneamente, la transforma y luego la envía a su "escondite" favorito (como Elasticsearch). Los datos a menudo se encuentran dispersos o en silos en muchos sistemas en muchos formatos. Logstash admite una variedad de entradas que extraen eventos de una multitud de fuentes comunes, todas al mismo tiempo. Logstash tiene más de 200 complementos.



Kibana es una plataforma de análisis y visualización de código abierto diseñada para trabajar con Elasticsearch. Con Kibana podemos buscar, ver e interactuar con los datos almacenados en los índices de Elasticsearch. Podemos realizar fácilmente análisis de datos avanzados y visualizar sus datos en una variedad de tablas, tablas y mapas. Kibana facilita la comprensión de grandes volúmenes de datos. Su sencilla interfaz basada en navegador nos permite crear y compartir rápidamente cuadros de mando dinámicos que muestran cambios en las consultas de Elasticsearch en tiempo real.



Beats es una familia de agentes muy ligeros que recolectan datos de los hosts y los envían a Elasticsearch o Logstash.

Todo comenzó con Packetbeat el cual tomaba datos acerca del tráfico de red. Después se añadieron 5 Elastic Beats de manera adicional para tratar: logs, métricas, datos de auditoría de Linux, información de los Windows Event Logs y datos de monitorización acerca de la disponibilidad de servicios, ampliando la familia Beats.

Además, existen más de 40 Beats creados por la comunidad para todo tipo de data operacional.

Durante el transcurso de la investigación sobre Kibana, fue constante el encontrarse menciones a Grafana, ¿Pero qué es Grafana? Daremos respuesta a esta pregunta en la siguiente sección.



Grafana

Grafana¹³ es una herramienta de visualización de código abierto (Licencia de Apache 2.0) escrita en lenguaje Go y con un HTTP API completo; que se puede usar sobre una variedad de diferentes almacenes de datos, como:

Graphite, InfluxDB, y también Elasticsearch.

Contiene un analizador que permite una fácil edición de métricas y funciones. Los usuarios pueden crear gráficos complejos de manera rápida.



Llegados hasta este punto podríamos pensar que Grafana es parecido a Kibana, pero lejos de esa afirmación, encontramos que realmente son bastante diferentes. Puntualizaremos las principales diferencias a continuación:

Kibana V/s Grafana

- Registros V/S métricas
 - La diferencia clave entre las dos herramientas de visualización proviene de su propósito. Grafana está diseñado para analizar y visualizar métricas como la CPU del sistema, la memoria, el disco y la utilización de E / S.
 - Grafana no permite consultas de datos de texto completo.
 - Kibana, por otro lado, se ejecuta sobre Elasticsearch y se usa principalmente para analizar mensajes de registro.
- Fuentes de datos e integraciones.

¹³ Sitio web de Grafana: <https://grafana.com/>

- Grafana fue diseñada para funcionar como una interfaz de usuario para analizar métricas. Como tal, puede funcionar con múltiples almacenes de datos de series de tiempo, incluyendo integraciones integradas con Graphite, Prometheus, InfluxDB, MySQL, PostgreSQL y Elasticsearch, y fuentes de datos adicionales utilizando complementos. Para cada fuente de datos, Grafana tiene un editor de consultas específico que se personaliza para las características y capacidades que se incluyen en esa fuente de datos.
- Kibana, por otro lado, está diseñado para funcionar solo con Elasticsearch y, por lo tanto, no admite ningún otro tipo de fuente de datos.
- Consultando
 - Consultar y buscar registros es una de las funciones más potentes de Kibana. Usando la sintaxis de Lucene, el DSL de Elasticsearch Query o el Kuery experimental, los datos almacenados en los índices de Elasticsearch se pueden buscar con los resultados mostrados en el área de visualización del registro principal en orden cronológico.
 - Con Grafana, los usuarios utilizan lo que se llama un Editor de consultas para realizar consultas. Cada fuente de datos tiene un Editor de consultas diferente adaptado para la fuente de datos específica, lo que significa que la sintaxis utilizada varía según la fuente de datos. Las consultas de grafito serán diferentes a las consultas de Prometheus, por ejemplo.

Hemos dejado claras las diferencias tanto conceptuales como de finalidad de cada herramienta, pero no podemos terminar este análisis sin mencionar a Prometheus. Pues, si comentamos anteriormente que al investigar sobre Kibana, fue inevitable leer acerca de Grafana, resultó que al investigar sobre Grafana fue inevitable leer acerca de Prometheus.



Prometheus

El proyecto está escrito en Go y licenciado bajo la Licencia Apache 2.

Prometheus¹⁴ es una BD de series de tiempo y un sistema de monitoreo y alertas. Las series de tiempo almacenan datos ordenados cronológicamente, midiendo variables a lo largo del tiempo y las bases de datos enfocadas a series de tiempo son especialmente eficientes en almacenar y consultar estos datos.

¹⁴ Sitio web de Prometheus: <https://prometheus.io/>

El proyecto tiene una comunidad muy activa de usuarios y desarrolladores. Su origen se remonta al 2012 en la compañía SoundCloud¹⁵, y desde entonces ha evolucionado a convertirse en un proyecto independiente de código abierto siendo parte de la Cloud Native Computing Foundation desde 2016, como el segundo proyecto alojado, después de Kubernetes.

Es una potente y ligera herramienta para recopilar y procesar métricas. Cuenta con un sistema propio de visualización de éstas métricas y con un sistema de envío de notificaciones. Es muy común conectarlo con Grafana, para mejorar la experiencia de generar gráficas a través de métricas recopiladas por Prometheus .

Pero al igual, que Grafana, cada uno en su campo, es una herramienta orientada a métricas, y como bien indican en el FAQ¹⁶ de Prometheus:

“Prometheus es un sistema para recopilar y procesar métricas, no un sistema de registro de eventos. “Usa algo como ELK Stack en su lugar.”

1.5. Justificación y estudio de la solución elegida.

Conclusiones sobre las herramientas

Las conclusiones que sacamos de las siete herramientas analizadas son:

Para monitorizar el estado de sistemas y/o servicios, elegiríamos claramente “Check_MK” por la facilidad de implementar plugins customizados, escritos en Python, además de por la ligereza tanto del núcleo como de sus agentes. (EJ: Monitorizar que el espacio libre de un Sistema de ficheros no sea inferior al 10% de su capacidad total)

Para monitorizar en profundidad las métricas de un servicio, elegiríamos “Prometheus + Grafana” (Ej: métricas de los frontales de un servicio web)

En un entorno en producción haríamos uso de un sistema híbrido, Check_MK + Prometheus + Grafana. Check_MK para recibir alertas de cualquier evento que consideremos crítico, y Grafana para visualizar las métricas de Prometheus y también de algunas alertas de Check_MK.

¹⁵ Prometheus en SoundCloud:

<https://developers.soundcloud.com/blog/prometheus-monitoring-at-soundcloud>

¹⁶ FAQ de Prometheus:

<https://prometheus.io/docs/introduction/faq/#how-to-feed-logs-into-prometheus>

Ahora bien, en el departamento de informática del IES Fernando Aguilar Quignon tenemos unas necesidades concretas bien distintas a las que pueda tener un servicio en producción. Lo más interesante, además de tener controladas las métricas de CPU, RAM, HDD, etc, de cada equipo del departamento, sería la centralización de logs, de manera que fuera posible sacar conclusiones de los cientos de datos que se generan diariamente en las aulas.

Conclusiones tales como:

- ¿En qué horas hay mayores conexiones de alumnos?
- ¿Qué alumnos se han conectado, en qué equipo y a qué hora?
- ¿Por dónde están navegando los alumnos?
- ¿Qué procesos están en ejecución en los equipos y cuántos recursos consumen?

O por otra parte, conocer qué recursos son los que realmente están siendo utilizados por los alumnos, para poder realizar una planificación correcta sobre qué ampliar en un futuro ¿Es la CPU o es la RAM lo que más limita a los alumnos? ¿Para un mismo entorno, qué consume más una implantación basada en Docker o en Máquinas Virtuales?

En base a toda la investigación aportada sólo hay una herramienta que de una solución práctica a todas estas necesidades, y que además sea de código abierto, ésta es:

ELK Stack, aunque con ciertos matices.



Extensión X-Pack

ELK Stack es de código abierto y gratuito a medias, porque hay ciertas funciones que sólo nos ofrecerán pagando una extensión llamada X-pack la cual añade:

- Seguridad y privacidad de la información.
- Monitorización del clúster.
- Sistema de alertas.
- Reportes gráficos sobre la conectividad.
- Machine learning.

La batalla elástica

A continuación detallaremos unos determinantes hechos que influyen directamente en la elección final de la herramienta que se terminará implantando en el departamento.

Antes de la versión 6.3 de Elasticsearch, X-pack era de código cerrado, y no se incluía en la instalación de Elasticsearch.

A partir ésta versión 6.3, el código de X-Pack estará abierto¹⁷ bajo la Licencia Elastic . Sin embargo, no será de "código abierto" ya que no estará cubierto por una licencia aprobada por OSI.

Desde éste momento X-pack no es una extensión que haya que instalar externamente, sino que desde ese momento Elastic ofrecería Elasticsearch en dos paquetes diferentes.

- La versión estándar:
 - Incluye todas las funciones de pago (X-Pack) que se pueden usar durante un tiempo determinado gracias a una versión de prueba.
- La versión OSS:
 - Estos paquetes, que están marcados con OSS (Open Source Software), solo contienen componentes libres que han sido publicados bajo la licencia Apache 2.0. (excluyendo todas las funcionalidades de X-pack)

Durante todo este tiempo, se había hecho muy popular un servicio en AWS llamado “Amazon Elasticsearch Service”, un SaaS introducido en 2015, el cual facilitaba la implementación, el uso y el escalado de los clústeres de Elasticsearch en la nube de AWS, pero que sufría grandes carencias por no disponer de las funcionalidades avanzadas de Elastic, ya que “Amazon Elasticsearch Service” no estaba y no está apoyado por Elastic.

Elastic tiene su propio SaaS, llamado: “Elastic Cloud¹⁸“ servicio iniciado tres años antes, en el 2012, e incluye todos los complementos comerciales de Elastic, y las últimas actualizaciones de todas sus herramientas.

Debido a la incompatibilidad por licencia de X-Pack, AWS no podía incluir éstas funcionalidades extras como parte de su “Amazon Elasticsearch Service” y ésto era objeto de críticas, tanto de usuarios que querían poder tener seguridad y privacidad de la información en su clúster de Elasticsearch y por otro lado, de consultores de

¹⁷ Comunicado de apertura de código:

<https://www.elastic.co/es/blog/opening-x-pack-phase-1-complete>

¹⁸ Sitio web de la elastic cloud: <https://www.elastic.co/es/cloud/>

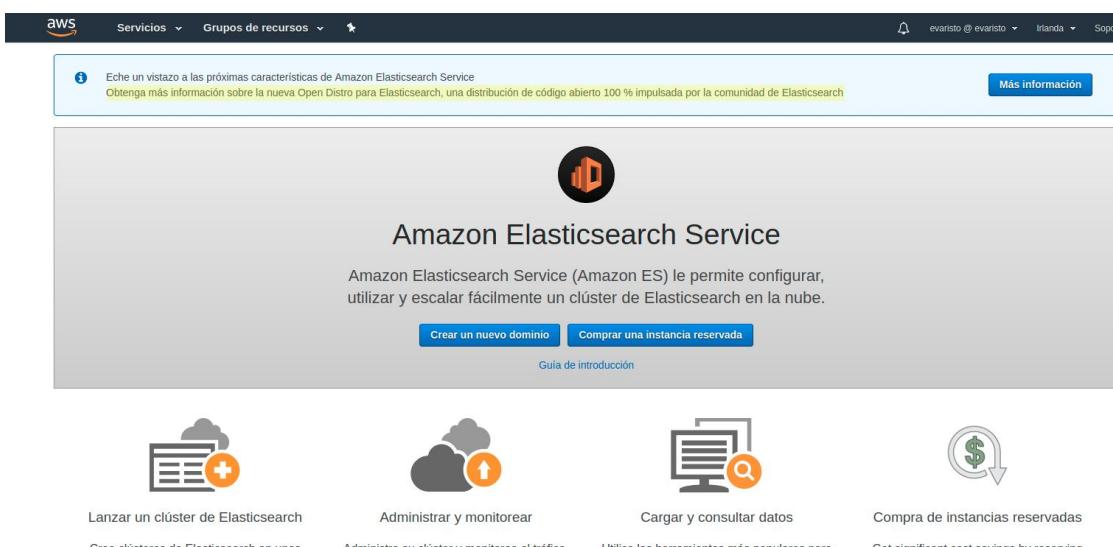
elastic que cargaban¹⁹, contra el servicio de AWS, e insistían en sus diferencias y debilidades.²⁰



Open Distro for Elasticsearch (AWS, Netflix y Expedia)

Es aquí cuando AWS, Netflix y Expedia Group unen sus fuerzas para desarrollar “Open Distro for Elasticsearch” un sustituto de X-Pack pero 100% de código abierto y gratis.

Concretamente el día 11 de marzo de 2019, es lanzado Open Distro for Elasticsearch, dos semanas previas al inicio de nuestra investigación. El lanzamiento se hace eco en el blog de AWS²¹ ese mismo día y dejan una nota en el panel de control de “Amazon Elasticsearch Service”.



The screenshot shows the AWS Elasticsearch Service landing page. At the top, there's a banner with a message about the new Open Distro distribution. Below the banner, the title "Amazon Elasticsearch Service" is displayed with a central icon. A brief description follows, mentioning its features like configuration, scaling, and monitoring. There are two main buttons: "Crear un nuevo dominio" and "Comprar una instancia reservada". Below these buttons is a "Guía de introducción". At the bottom of the main section, there are four icons with corresponding labels: "Lanzar un clúster de Elasticsearch" (Create a new Elasticsearch cluster), "Administrar y monitorear" (Manage and monitor), "Cargar y consultar datos" (Load and query data), and "Compra de instancias reservadas" (Buy reserved instances). Each label has a small explanatory text underneath it.

La versión más reciente que ofrecía Amazon Elasticsearch Service hasta la fecha era la 6.5, era un poco extraño que se hubieran quedado “atascados” en esa versión, cuando Elasticsearch ya iba por la versión 6.7, 7.1.. etc... y ahora todo cobraba sentido, AWS había estado inmerso en el lanzamiento de OpenDistro, por

¹⁹ Consultores cargan contra AWS

<https://code972.com/blog/2017/12/111-why-you-shouldnt-use-aws-elasticsearch-service>

²⁰ Diferencias entre Elastic Cloud y AWS Elasticsearch:

<https://www.elastic.co/es/blog/hosted-elasticsearch-services-roundup-elastic-cloud-and-amazon-elasticsearch-service>

²¹ Lanzamiento de Open Distro for Elasticsearch:

<https://aws.amazon.com/es/blogs/opensource/keeping-open-source-open-open-distro-for-elasticsearch/> || <https://aws.amazon.com/es/blogs/aws/new-open-distro-for-elasticsearch/>

lo que no estaba en sus planes seguir actualizando las versiones de Amazon Elasticsearch Service.

Como podemos observar en el siguiente extracto, a Elastic no le sentó nada bien esta maniobra:

No confunda el Open Distro de AWS con Elasticsearch con el altruismo²²

(...) Este es un movimiento retaliatorio de AWS, debido a un movimiento reciente de Elastic para colocar X-Pack (las adiciones "empresariales" a Elasticsearch) bajo una licencia que restringe a Amazon para que lo use. Debido a que AWS Elasticsearch es famoso por ser el servicio de más rápido crecimiento de AWS, Amazon no quiere perder su base de clientes o quedarse atrás con características importantes, por lo que optaron por escribir su reemplazo de X-Pack. Y también decidieron lanzarlo bajo la licencia Apache 2, (...)

OpenDistro para Elasticsearch es solo una manera de que AWS conserve algunos clusters de AWS Elasticsearch y no los pierda con el X-Pack de Elastic.

La afirmación de AWS de altruismo en este caso no es más que hipocresía.

(...)

Dejando aparte las guerras comerciales, nosotros recibimos con los brazos abiertos esta alternativa. Veamos a continuación qué ofrece Opendistro²³ exactamente:

Seguridad

Open Distro para ElasticSearch es compatible con OpenSSL y TLS 1.2, lo que permite cumplir con estrictos requisitos de seguridad.

Open Distro para ElasticSearch puede aprovechar las infraestructuras de autenticación existentes como LDAP / Active Directory, SAML, Kerberos, tokens web JSON, certificados TLS y autenticación Proxy / SSO para la autenticación del usuario.



²² Críticas a Opendistro:

<https://code972.com/blog/2019/03/116-dont-confuse-awss-open-distro-for-elasticsearch-with-altruism>

²³ Sítio web de OpenDistro for Elasticsearch: <https://opendistro.github.io/for-elasticsearch/>

Es compatible con el control de acceso basado en roles. Los roles controlan las operaciones del clúster, el acceso a los índices e incluso los campos y documentos a los que pueden acceder los usuarios.

También es compatible con entornos de múltiples inquilinos, lo que permite que varios equipos comparten el mismo clúster y solo puedan acceder a los datos y paneles de su equipo.

Alertas

Open Distro For ElasticSearch admite alertas en condiciones definidas.

Proporciona herramientas basadas en UI + API para crear monitores interactivamente.

La notificaciones pueden ser mediante Webhooks, SMTP(próximamente) y Slack.

Las alertas se pueden ver utilizando simples paneles de control.

Consulta con SQL

Open Distro para ElasticSearch permite extraer/analizar datos usando una sintaxis SQL familiar.

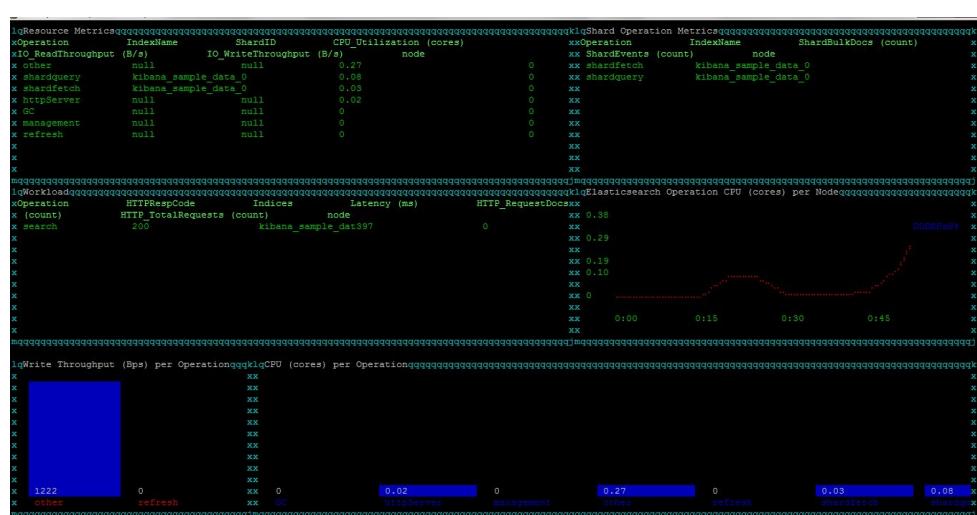
Los datos se pueden extraer en formato de documento CSV o JSON.

También proporciona el controlador JDBC para que se pueda conectar con una amplia gama de herramientas de terceros.

Analizador de rendimiento

Performance Analyzer es una herramienta que permite obtener métricas de rendimiento de ElasticSearch como CPU, memoria, operaciones de E / S a través de la API REST.

También proporciona un panel de control de monitoreo basado en CLI que puede ser útil para monitorear muchos puntos de datos



Justificación final de la decisión

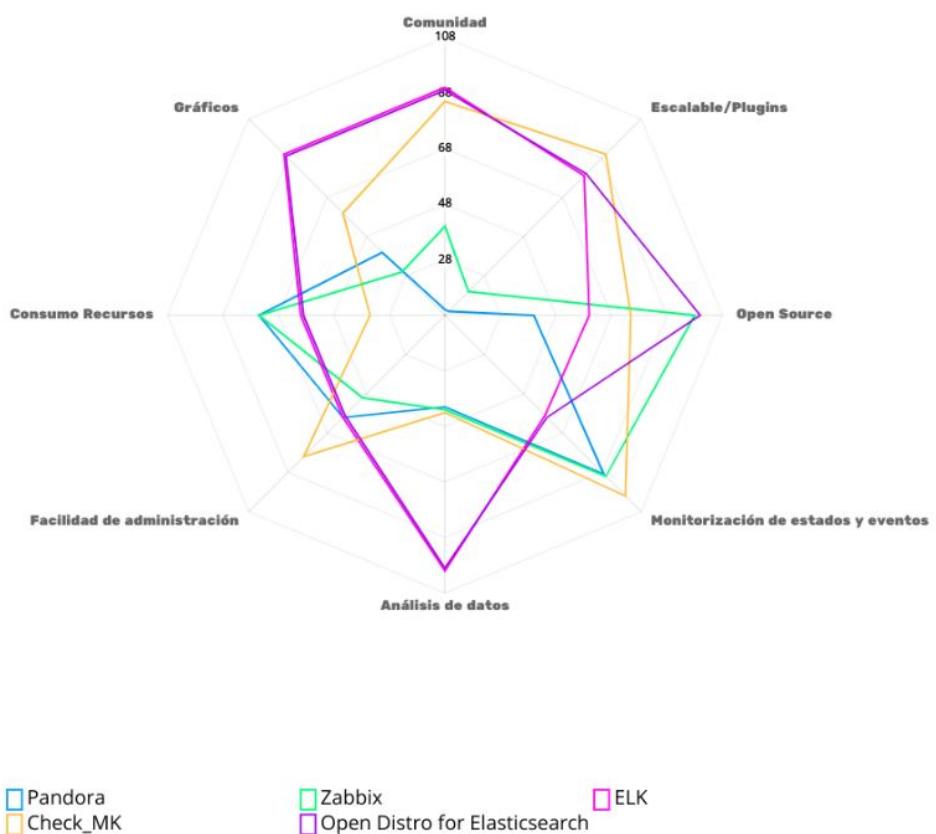
Echamos de menos una función Machine learning, pero no descartamos verla en un futuro próximo, ya que Opendistro for Elasticsearch está en continua evolución.

La promesa inicial de “Opendistro for Elasticsearch” es ir dos semanas por detrás de la última versión que saque Elastic, aunque ahora mismo se están retrasando un poco, ya que la versión 7.x de Elastic trae muchísimas novedades consigo.

Version history

Open Distro for Elasticsearch version	Release highlights	Release date	Elasticsearch version
0.9.0	Bumps Elasticsearch version.	1 May 2019	6.7.1
0.8.0	Bumps Elasticsearch version.	5 April 2019	6.6.2
0.7.1	Fixes Kibana multitenancy.	29 March 2019	6.5.4
0.7.0	Initial release.	11 March 2019	6.5.4

Para concluir, se ha elaborado un gráfico que resume los puntos fuertes de cada herramienta:



En definitiva, nos encontramos con el problema de que no queremos que nuestros datos viajen en plano. Nos gusta la idea de tener seguridad de nivel empresarial a coste cero.

Hemos investigado las alternativas²⁴ de terceros, tanto de la comunidad como de empresas externas, que han implementado soluciones individuales para cada funcionalidad de X-pack.

Nos interesan concretamente las orientadas a la seguridad y protección de la información.

Encontramos que existen dos: Search Guard y ReadonlyREST.

Pero Search Guard no contempla LDAP en su versión gratuita y ReadonlyREST sólo ofrece gratuitamente su extensión para Elasticsearch pero no para Kibana.

Además la idea de integrar por nuestra cuenta éstas herramientas entraña varios problemas, si nos paramos a pensar en las actualizaciones venideras, que no serán pocas conociendo el ritmo de actualización de Elastic.

Así que teniendo la suerte de contar con el proyecto, Opendistro for Elasticsearch, y tras confirmar lo activa que está su comunidad, apostamos por éste.

Antes de terminar anotamos que curiosamente las características de seguridad de “Opendistro for Elasticsearch” incluyen contribuciones²⁵ de Floragunn (Los creadores de Search Guard)

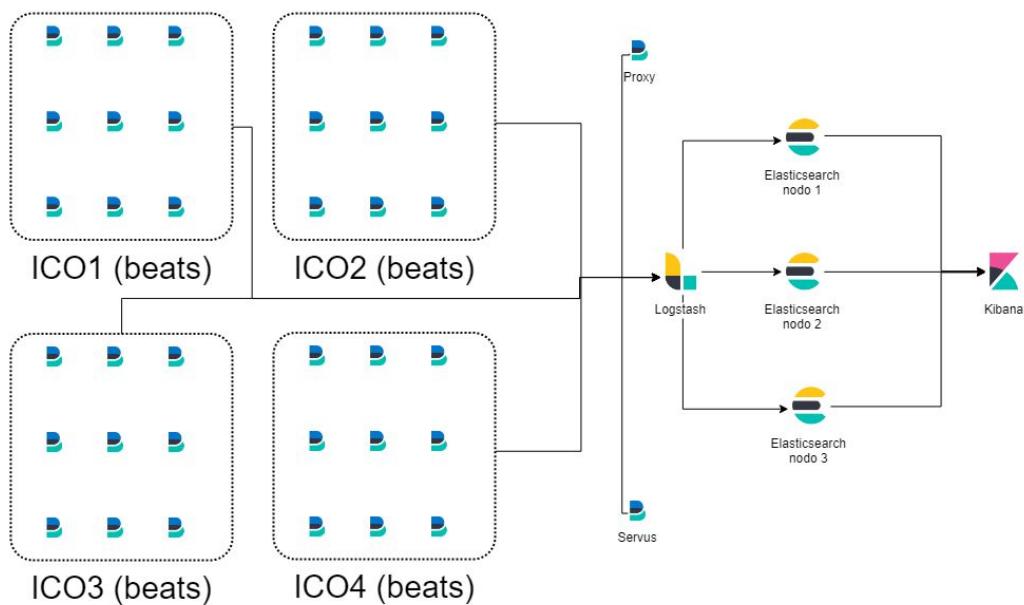
Todo ésto eran los matices que comentábamos. En base a lo descrito anteriormente, no queda duda, la opción elegida para implantar en el departamento de informática del IES Fernando Aguilar Quignon es: ELK Stack con “Opendistro for Elasticsearch”.

²⁴ En el Anexo 1 (Alternativas de código abierto y gratis a X-Pack por componente) se detallan las alternativas que existen actualmente.

²⁵ Tema de la comunidad Opendistro donde se confirma la contribución de Floragunn:
<https://discuss.opendistrocommunity.dev/t/why-no-attribution-to-searchguard/131>

1.6. Modelado de la solución

Si se desea alta disponibilidad y tolerancia a fallos, se recomienda crear un clúster de Elasticsearch de tres nodos. Elasticsearch es altamente escalable horizontalmente, por lo que se podrá comenzar con menos nodos e ir ampliéndolo según se requiera.



IES Fernando Aguilar Quignon



Evaristo R. Rivieccio Vega

En entornos muy grandes de producción se recomiendan dos nodos de Logstash tras un balanceador, pero en el caso del departamento con un nodo será suficiente. Elasticsearch recomienda que la memoria asignada al proceso de Elasticsearch esté cerca de la mitad de la memoria total disponible.

En las pruebas que se han hecho, se ha estado corriendo Elasticsearch con el montón JVM limitado a 4GB y el rendimiento ha sido aceptable. En todo momento Elasticsearch ha estado consumiendo siempre el máximo asignado, 4GB.

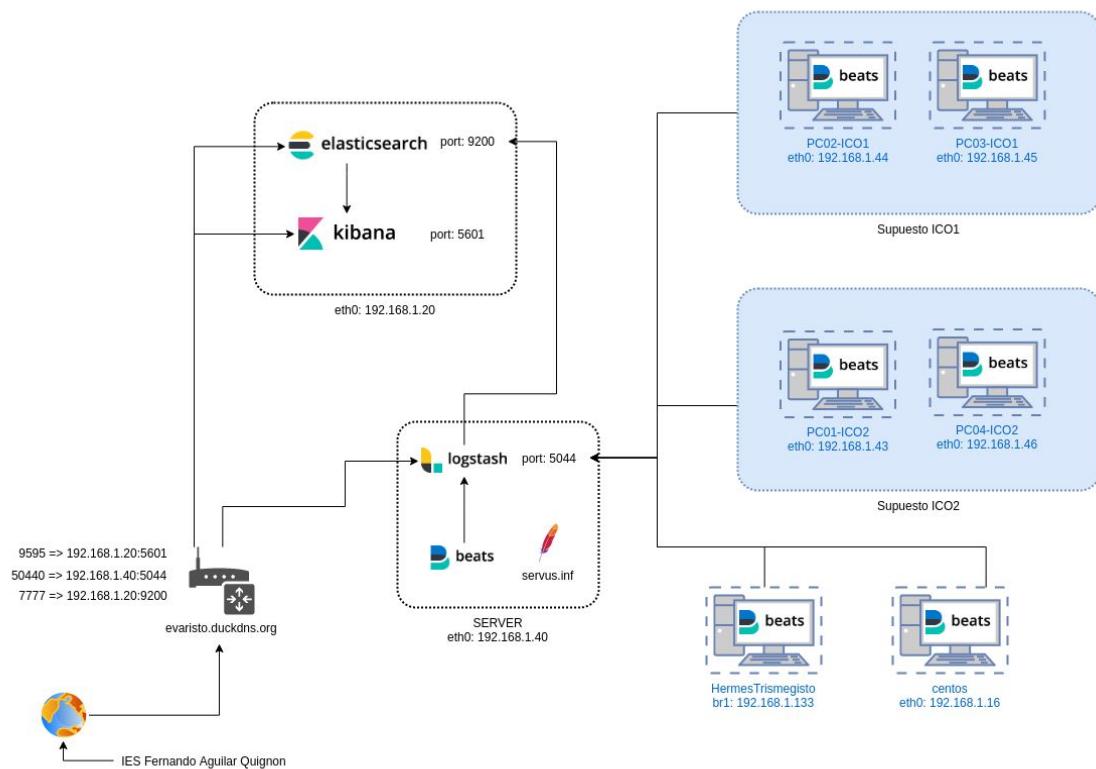
Logstash por su parte nunca ha consumido más de 2GB pese a que tenía de margen 4GB igualmente.

Kibana no consume casi recursos, menos de 1 GB en todo caso.

Los Beats son los agentes extremadamente ligeros, que correrán en cada equipo que se quiera monitorizar.

1.6.1. Recursos utilizados

Para el estudio del proyecto se han utilizado máquinas virtuales, corriendo en el equipo personal del autor de ésta investigación ("entorno de tests" de aquí en adelante), simulando el entorno del departamento, garantizando en todo momento los requisitos de seguridad y de rendimiento óptimos de los sistemas, los cuales no se han visto afectados por el sistema de monitorización, así como la red, que tampoco se ha visto afectada por el tráfico de logs en tiempo real.



- Anfitrión “HermesTrismegisto” con 64 GB de RAM
 - Con agentes Beats.
 - Una MV de 8 GB de RAM con Elasticsearch y Kibana
 - Montón JVM de Elasticsearch limitado a 4GB
 - Una MV de 8 GB de RAM con Logstash y una imitación de servus.inf
 - Montón JVM de Logstash limitado a 4GB
 - Una MV centos de 4 GB
 - Con agentes Beats.
 - Cuatro MV Ubuntu de 2 GB cada una simulando cuatro PCs del departamento
 - Con agentes Beats

Como Hipervisor se ha utilizado QUEM/KVM mediante Virt-Manager.

Además se ha preparado la infraestructura de red para permitir recibir conexiones del exterior tanto para Logstash como para Elasticsearch, con la idea de, en la fase pruebas, recibir directamente los datos del departamento en el “entorno de tests”, como detallaremos en la sección “3. Fase de pruebas”.

1.7. Planificación temporal

Semana del 18 de marzo: investigación de las distintas herramientas disponibles en el mercado y pruebas de las mismas.

Semana del 25 de marzo a 12 de abril: estudio de las opciones elegidas (Ver curso en OpenWebinars de ELK) integraciones, etc...

15 de abril a 26 de abril: Despliegue en “entorno de test” del laboratorio y estudio de la arquitectura.

29 de abril al 10 de mayo: Estudio en profundidad de la herramienta. Tratamiento de datos.

13 de mayo al 31 de mayo: Perfeccionamiento de la solución en base a sugerencias recogidas del tutor de seguimiento del proyecto.

01 de junio al 13 de junio: Maquetado de la documentación.

14 de junio: Entrega de la documentación.

15 de junio al 20 de junio: Elaboración de la presentación y preparación de la exposición.

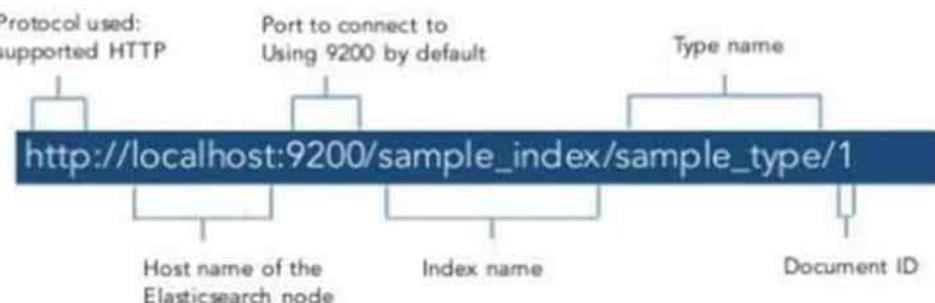
21 de junio: Exposición ante tribunal de profesores.

2. Arquitectura de los componentes principales

2.1 Arquitectura de Elasticsearch

Principales características de Elasticsearch

- Su escalabilidad horizontal, es su principal característica y en la que se centró el diseño de su arquitectura.
- Tiene respuestas próximas al tiempo real, ya que la latencia es menor de un segundo desde que se indexa hasta que el dato está disponible para la búsqueda.
- Es tolerante a fallos en los nodos, lo que ofrece alta disponibilidad al tener replicación de los datos en nodos diferentes.
- Dispone de funciones de búsqueda en texto completo, ya que se considera todo el contenido de los documentos para la búsqueda, a diferencia de lo que ocurre en una base de datos convencional, en la que solo se consideran campos concretos como el título o las referencias.
- Está orientado a documentos JSON.
- No requiere esquemas, por lo que en un indexado rápido se autodetectan los tipos de cada campo al generarlos.
- Tiene un desarrollo amigable con unas APIs sencillas.



Escalabilidad de Elasticsearch

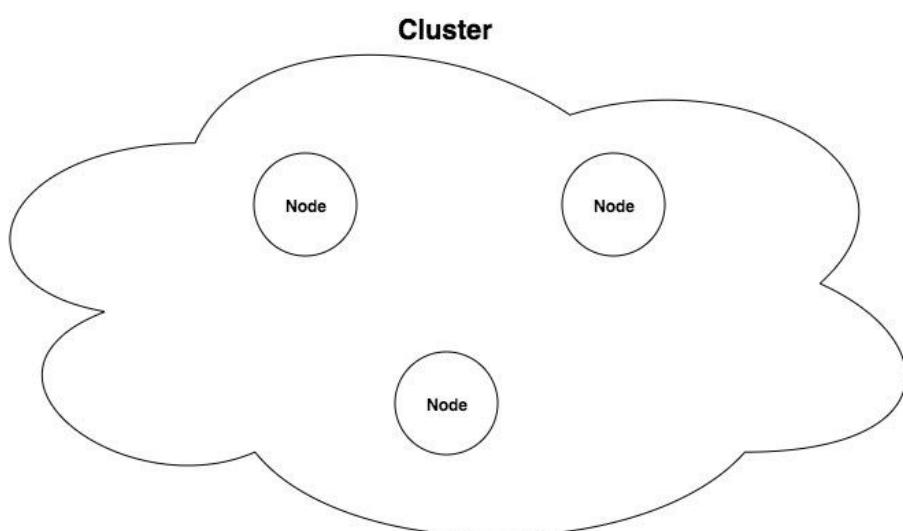
- Elasticsearch tiene una escalabilidad horizontal en lugar de una escalabilidad vertical.
 - La diferencia entre ambas es que la escalabilidad vertical implica tener un mejor hardware, es decir, cuando necesitamos tener un sistema más potente lo que hacemos es comprar más disco duro, memoria o procesador. Esto es limitado, ya que la cantidad de mejoras que podemos hacer en un mismo equipo siempre es limitada, y llegará un momento en el que no se pueda ampliar más.

- En la escalabilidad horizontal se tienen varios nodos trabajando como único servidor, con lo cual se puede ampliar indefinidamente.

Clusters y Nodos

Los clústeres son una colección de nodos que se comunican entre sí para leer y escribir en un índice. Un clúster necesita un nombre único para evitar que se unan nodos innecesarios.

Un nodo es una instancia única de Elasticsearch. Por lo general, se ejecuta una instancia por máquina. Se comunican entre sí mediante llamadas de red para compartir la responsabilidad de leer y escribir datos. Un nodo maestro organiza todo el cluster.



Cómo funciona el almacenamiento en Elasticsearch

- Almacena documentos en índices, que son la unidad principal. Antes de Elasticsearch 6 se solía utilizar la analogía entre un índice y una base de datos, y una tabla y un documento, y era posible para un índice de Elasticsearch tener varios tipos. Sin embargo se comprobó que este sistema generaba problemas en la capacidad de comprensión de Lucene, que está por debajo de Elasticsearch, y se decidió que sólo fuera posible tener un solo tipo de datos por índice.
- Los índices se dividen en shards (fragmentos)
- Cada fragmento puede ubicarse en un nodo diferente del clúster.
- Tiene tolerancia a fallos, lo que se consigue mediante shards primarios y réplicas de los mismos en otros nodos.
- Las escrituras se realizan sobre shards primarios, los cuales luego son replicados.

- Las lecturas se pueden realizar tanto sobre shards primarios como sobre las réplicas, con lo cual mejora la capacidad de lectura.

Ejemplo del funcionamiento de Elasticsearch

- Ejemplo: índice dividido en 2 shards (P0 y P1). 3 nodos y 2 réplicas por shard



- Si falla un nodo: réplicas pasan a ser primarias.



En la parte superior de la imagen podemos ver que tenemos un índice dividido en dos shards, denominados P0 y P1, con tres nodos y dos réplicas por cada shard.

Tenemos el nodo 1, el nodo 2 y el nodo 3, y dentro de ellos marcadas en verde los shards primarios y en gris las réplicas.

Si falla uno de los nodos, las réplicas pasan a ser primarias, y al tener dos shards menos se volverían a generar nuevas réplicas.

En este caso nos quedaría como vemos en la parte inferior de la imagen, en la que habría fallado el nodo 1 maestro, pero no ha ocurrido nada porque el rol de maestro lo ha pasado ocupar el nodo 2, y el sistema sigue funcionando como antes.

Otras características de los shards y las réplicas

- El número de shards se puede definir al crear el índice y es fijo una vez que se ha creado ese índice.
- El número de réplicas se puede cambiar dinámicamente.
- Por defecto, un índice tiene cinco shards y una réplica, la cual solo tiene efectos si tenemos dos nodos, ya que si solo tenemos uno, no habría ningún sitio donde se pueda replicar.



Lucene

Lucene es el nombre del motor de búsqueda que alimenta a Elasticsearch. Es un proyecto de código abierto de la Fundación Apache. No hay necesidad de interactuar con Lucene directamente, al menos la mayor parte del tiempo, cuando se ejecuta Elasticsearch. Pero hay algunas cosas importantes que debemos conocer.

Segmentos de lucene

Elasticsearch usa Apache Lucene , una biblioteca de búsqueda de texto completo escrita en Java y desarrollada por Doug Cutting (creador de Apache Hadoop), que internamente usa una estructura de datos llamada índice invertido diseñado para servir resultados de búsqueda de baja latencia.

Un documento es la unidad de datos en Elasticsearch y se crea un índice invertido al tokenizar los términos en el documento, creando una lista ordenada de todos los términos únicos y asociando una lista de documentos con el lugar donde se puede encontrar la palabra

Es muy similar a un índice en la parte posterior de un libro que contiene todas las palabras únicas en el libro y una lista de páginas donde podemos encontrar esa palabra. Cuando decimos que un documento está indexado, nos referimos al índice invertido.

Veamos cómo se ve el índice invertido para los siguientes dos documentos:

- Doc 1: Programa Insight Data Engineering Fellows
- Doc 2: Programa Insight Data Science Fellows

Token	Documents
data	Doc 1, Doc 2
engineering	Doc 1
fellows	Doc 1, Doc 2
insight	Doc 1, Doc 2
program	Doc 1, Doc 2
science	Doc 2

Si queremos encontrar documentos que contengan el término "insight", podemos escanear el índice invertido (donde se clasifican las palabras), buscar la palabra "Insight" y devolver las identificaciones del documento que contienen esta palabra, que en este caso sería Doc. 1 y Doc 2.

Para mejorar la capacidad de búsqueda (p. Ej., Para obtener los mismos resultados tanto en minúsculas como en mayúsculas), los documentos se analizan primero y luego se indexan.

Cada índice de Elasticsearch se divide en fragmentos. Los fragmentos son la división de un índice. Cada fragmento de Elasticsearch es un índice de Lucene. El número máximo de documentos que puede tener en un índice de Lucene es de 2.147.483.519. El índice de Lucene se divide en archivos más pequeños llamados segmentos. Un segmento es un pequeño índice de Lucene. Lucene busca en todos los segmentos secuencialmente.

Elasticsearch Index							
Elasticsearch shard		Elasticsearch shard		Elasticsearch shard		Elasticsearch shard	
Lucene index		Lucene index		Lucene index		Lucene index	
Segment	Segment	Segment	Segment	Segment	Segment	Segment	Segment

De vez en cuando, Lucene fusiona segmentos más pequeños en uno más grande. La fusión también se puede activar manualmente desde la API de Elasticsearch.

Cuantos más segmentos tenemos, más lenta será la búsqueda. Esto se debe a que Lucene tiene que buscar en todos los segmentos en secuencia, no en paralelo. Tener un pequeño número de segmentos mejora los resultados de búsqueda.

Lucene no borra; actualiza.

Lucene realiza una copia en escritura al actualizar y eliminar un documento. Esto significa que el documento nunca se elimina del índice. En su lugar, Lucene marca el documento como eliminado y crea otro cuando se activa una actualización.

Esta copia sobre escritura tiene una consecuencia operativa. A medida que se actualice o se eliminen documentos, los índices crecerán en el disco a menos que los eliminemos por completo. Una solución para eliminar realmente los documentos marcados es forzar la fusión de los segmentos de Lucene.

Durante una fusión, Lucene toma 2 segmentos y mueve el contenido a un tercero nuevo. A continuación, los segmentos antiguos se eliminan del disco. Significa que Lucene necesita suficiente espacio libre en el disco para crear un segmento del tamaño de ambos segmentos que necesita fusionar.

Tener muchos fragmentos puede ser bueno o terrible para un clúster. La administración de índices y fragmentos puede sobrecargar el nodo maestro, lo que podría dejar de responder, lo que podría provocar un comportamiento extraño y desagradable. Hay que Intentar no ejecutar más de 10.000 índices abiertos y 50.000 fragmentos primarios en el mismo clúster.

Tipos de nodos

Existen varios tipos de nodos:

Nodo maestro

Controla el clúster Elasticsearch y es responsable de todas las operaciones de todo el clúster, como crear/eliminar un índice, realizar un seguimiento de qué nodos forman parte del clúster y asignar fragmentos a los nodos. El nodo maestro procesa un estado de clúster a la vez y transmite el estado a todos los otros nodos que responden con confirmación al nodo maestro.

Se puede configurar un nodo para que se convierta en un nodo maestro configurando la propiedad `node.master` como verdadera (predeterminada) en `elasticsearch.yml`

Para grandes clústeres de producción, se recomienda tener un nodo maestro dedicado para controlar el clúster y no atender las solicitudes de los usuarios.

Nodo de datos

Contiene los datos y el índice invertido. De forma predeterminada, cada nodo está configurado para ser un nodo de datos y la propiedad node.data se establece en true en elasticsearch.yml.

Si desea tener un nodo maestro dedicado, cambiaremos la propiedad node.data a falso.

Nodo Cliente

Si establecemos node.master y node.data en falso, entonces el nodo se configura como un nodo cliente y actúa como un equilibrador de carga que enruta las solicitudes entrantes a diferentes nodos en el clúster.

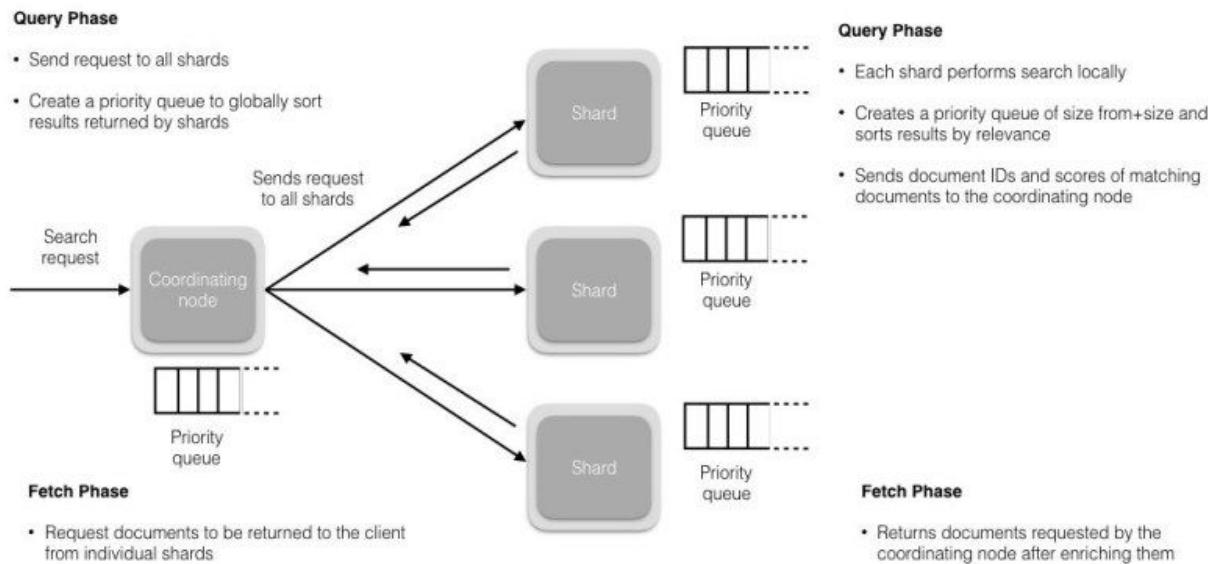
Nodo coordinador

El nodo en el clúster de Elasticsearch con el que se conecta como cliente se denomina nodo coordinador. El nodo coordinador encamina las solicitudes del cliente al fragmento apropiado en el clúster. Para las solicitudes de lectura, el nodo coordinador selecciona un fragmento diferente cada vez que sirve la solicitud para equilibrar la carga.

Flujo de consultas de búsqueda en Elasticsearch

La búsqueda tiene 2 fases principales:

- Fase de consulta:
 - Una solicitud de búsqueda primero llega a un nodo de coordinación y reenviará la consulta a una copia (primaria o réplica) de cada fragmento en el índice.
 - Cada fragmento ejecutará la consulta localmente y entregará las ID de documento de los resultados más relevantes (por defecto, 10) al nodo coordinador, que a su vez se fusionará y ordenará para encontrar las ID de documento de los mejores resultados globales relevantes.
- Fase de búsqueda:
 - Después de que el nodo de coordinación ordena todos los resultados para generar una lista de documentos ordenada globalmente, solicita los documentos originales de todos los fragmentos. Todos los fragmentos enriquecen los documentos y los devuelven al nodo coordinador. Finalmente, el resultado final de la búsqueda se envía de vuelta al cliente.



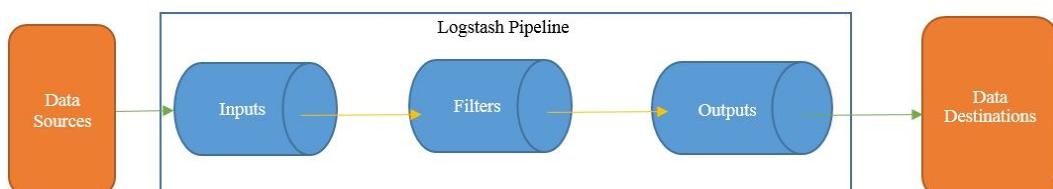
2.2 Arquitectura de Logstash

Logstash fue desarrollado originalmente por Jordan Sissel para manejar la transmisión de una gran cantidad de datos de registro de múltiples fuentes, y después de que Sissel se unió al equipo de Elasticsearch (entonces llamado Elasticsearch), Logstash evolucionó de una herramienta independiente a una parte integral de la Pila ELK (Elasticsearch, Logstash, Kibana).

Para poder implementar un sistema de registro centralizado efectivo, se requiere una herramienta que pueda extraer datos de múltiples fuentes de datos y darle un significado.

Esta es la función que desempeña Logstash: se encarga de las tareas de extracción y recepción de datos de varios sistemas, transformándolos en un conjunto significativo de campos y, finalmente, transmitiendo la salida a un destino definido para su almacenamiento

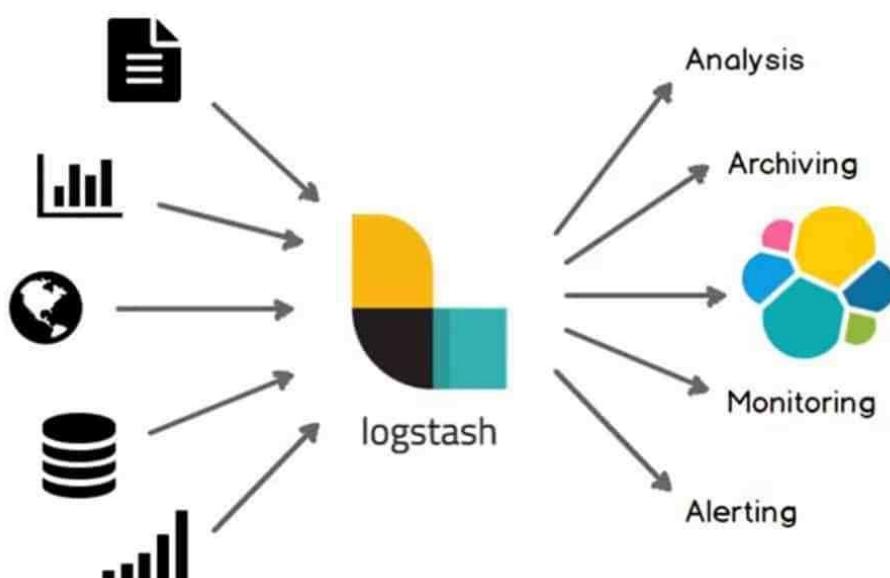
El proceso de procesamiento de eventos de Logstash tiene tres etapas, que son: Entradas, Filtros y Salidas. Una tubería de Logstash tiene dos elementos requeridos; entrada, salida, y, opcionalmente, filtros:



Las entradas crean eventos, los filtros modifican los eventos de entrada y las salidas los envían al destino. Las entradas y salidas son compatibles con los códecs que le permiten codificar o decodificar los datos a medida que entran o salen de la tubería sin tener que usar un filtro separado.

Logstash usa colas en memoria dentro de las etapas de la tubería de forma predeterminada (Entrada a filtro y Filtro a salida) para amortiguar eventos. Si Logstash termina de forma insegura, se perderán todos los eventos almacenados en la memoria. Para evitar la pérdida de datos, se puede habilitar Logstash para que persista los eventos en vuelo en el disco haciendo uso de colas persistentes.

Logstash: puede recopilar registros de una variedad de fuentes (usando complementos de entrada), procesar los datos en un formato común usando filtros y transmitir datos a una variedad de fuentes (usando complementos de salida). Se pueden encadenar múltiples filtros para analizar los datos en un formato común. Juntos, construyen una tubería de procesamiento de Logstash.



A continuación vemos una selección de “Input Plugins”, “Filter Plugins” y “Output Plugins”:

Input Plugins	Filter Plugins	Output Plugins
Beats	Aggregate	CSV
Elasticsearch	CSV	Elasticsearch
Kafka	Date	Email
Graphite	geoip	File
Heartbeat	grok	Graphite
Ttp	Json	Http
JDBC	sleep	Jira
File	urlencode	Kafka
Log4j	UUID	Nagios
Redis	xml	Redis
Stdin		Stdout
TCP		S3
Twitter		TCP
		UDP

2.3 Beats - Lumberjack

Lumberjack se desarrolló inicialmente como un experimento para subcontratar la tarea de extracción de datos y estaba destinado a ser utilizado como un cargador ligero para recopilar registros antes de enviarlos para su procesamiento en otra plataforma.

Escrito en Go, el concepto detrás de Lumberjack era desarrollar un protocolo de red que fuera más eficiente en el manejo de grandes volúmenes de datos, que tuviese poca huella de memoria y soporte para el cifrado.

Un giro dramático de los eventos llevó a Lumberjack a cambiar el nombre de Logstash-Forwarder , donde el primero ahora solo constituye el protocolo de red y el último es el programa de registro real.

El nacimiento de Beats

Se desarrolló una segunda versión del protocolo Lumberjack (extrañamente, sin documentación disponible en la web), en desuso de Logstash-Forwarder. Este nuevo protocolo se convirtió en la columna vertebral de una nueva familia de transportistas llamada Beats.

Las diferencias clave entre las dos versiones fueron la compatibilidad con el anidamiento JSON en un mensaje, un mejor manejo de la contrapresión con una reducción eficiente del tamaño de la ventana y la capacidad de reconocimiento en la mitad de la ventana.

El primer "beat" fue Packetbeat, cuyos desarrolladores se unieron a Elastic en mayo de 2015 , y Filebeat, anunciado como el "Forwarder Logstash de próxima generación", se presentó poco después.

Los diferentes Beats se utilizan como agentes ligeros instalados en los diferentes servidores de la infraestructura para enviar registros o métricas. Estos pueden ser ficheros de registro (Filebeat), métricas de la red (Packetbeat), métricas del servidor (Metricbeat), o cualquier otro tipo de datos que puedan ser recopilados por el creciente número de Beats desarrollados por Elastic y la comunidad.

Lumberjack-protocol (El protocolo del leñador)

Licencia Apache 2.0

El protocolo de leñador²⁶ está activamente en desarrollo en Elastic, sin embargo, la documentación del protocolo ha quedado obsoleta con respecto a la implementación real del proyecto Elastic Beats y la entrada de Logstash Beats. Aunque aún no se han documentado los cambios entre los protocolos v1 y v2, las necesidades que conducen a este protocolo son:

- Encriptación y autenticación para proteger.
- La compresión debe usarse para reducir el ancho de banda.
- La latencia de ida y vuelta no debe dañar el rendimiento
- Reconocimiento de mensajes a nivel de aplicación

El comportamiento de secuencia y acuse de recibo (incluida la ventana deslizante, etc.) es similar a TCP, pero en lugar de bytes, los mensajes son la unidad base.

Un escritor con un tamaño de ventana de 50 eventos puede enviar hasta 50 eventos sin apilar antes de bloquear. Un lector puede reconocer el "último evento" recibido para admitir un reconocimiento masivo.

El transporte de bytes ordenado y confiable se garantiza mediante el uso de TCP, y este protocolo tiene como objetivo proporcionar un transporte de mensajes confiable y de nivel de aplicación.

El Cifrado y la autenticación debe ser manejada por TLS.

Filebeat - Arquitectura

Filebeat consta de dos componentes principales: inputs y recolectores. Estos componentes trabajan juntos para seguir ficheros y enviar datos de eventos a la salida que especifique.

El funcionamiento en general de Filebeat es el siguiente: cuando se inicia Filebeat, se inician una o más entradas que buscan en las ubicaciones que se han especificado para los datos de registro, para cada registro que Filebeat localiza, Filebeat inicia un recolector. Cada recolector lee un registro único para el nuevo contenido y envía los nuevos datos de registro al framework libbeat²⁷, que agrega

²⁶ Más información acerca del protocolo del leñador:

<https://github.com/logstash-plugins/logstash-input-beats/blob/master/PROTOCOL.md>

²⁷ libbeat: <https://github.com/elastic/beats/tree/master/libbeat>

los eventos y envía los datos agregados a la salida que se ha configurado para Filebeat.

Profundicemos en éste tema:

¿Qué es un recolector?

Un recolector es responsable de leer el contenido de un solo fichero. El recolector lee cada fichero, línea por línea, y envía el contenido a la salida. Se inicia un recolector para cada fichero. El recolector es responsable de abrir y cerrar el fichero, lo que significa que el descriptor del fichero permanece abierto mientras el recolector está funcionando. Si un fichero se elimina o cambia de nombre mientras se está recolectando, Filebeat continúa leyendo el fichero. Esto tiene el efecto secundario de que el espacio en el disco está reservado hasta que el recolector se cierre. De forma predeterminada, Filebeat mantiene el fichero abierto hasta que se alcanza “close_inactive”.

Cerrar un recolector tiene las siguientes consecuencias:

- El manejador de ficheros se cierra, liberando los recursos subyacentes si el fichero se eliminó mientras el recolector seguía leyendo el fichero.
- La recolección del fichero solo se volverá a iniciar después de que haya transcurrido “scan_frequency”.
- Si el fichero se mueve o elimina mientras el recolector está cerrado, la recolección del fichero no continuará.

Para controlar cuándo se cierra un recolector, se usa la opción de configuración “close_”*

¿Qué es un input?

Un input es responsable de administrar los recolectores y encontrar todas las fuentes para leer.

Si el tipo de entrada es un registro, la entrada encuentra todos los ficheros en la unidad que coinciden con las rutas globales definidas e inicia un recolector para cada fichero. Cada entrada se ejecuta en su propia rutina Go. Filebeat actualmente admite varios tipos de entrada. Cada tipo de entrada se puede definir varias veces. La entrada de registro comprueba cada fichero para ver si es necesario iniciar un recolector, si ya se está ejecutando uno o si se puede ignorar el fichero.

Las nuevas líneas solo se seleccionan si el tamaño del fichero ha cambiado desde que se cerró el recolector.

¿Cómo mantiene Filebeat el estado de los ficheros?

Filebeat mantiene el estado de cada fichero y con frecuencia lo vacía en el disco en el fichero de registro. El estado se utiliza para recordar el último desplazamiento desde el cual un recolector estaba leyendo y para garantizar que se envíen todas las líneas de registro. Si no se puede acceder a la salida, como Elasticsearch o Logstash, Filebeat realiza un seguimiento de las últimas líneas enviadas y continuará leyendo los ficheros tan pronto como la salida vuelva a estar disponible.

Mientras Filebeat se está ejecutando, la información de estado también se guarda en la memoria para cada entrada. Cuando se reinicia Filebeat, los datos del fichero de registro se utilizan para reconstruir el estado, y Filebeat continúa cada recolector en la última posición conocida.

Para cada entrada, Filebeat mantiene un estado de cada fichero que encuentra. Debido a que los ficheros pueden ser renombrados o movidos, el nombre de fichero y la ruta no son suficientes para identificar un fichero. Para cada fichero, Filebeat almacena identificadores únicos para detectar si un fichero se recolectó previamente.

Es decir que en caso de problemas de red o interrupciones en las transmisiones, Filebeat recordará dónde se quedó cuando se restableció la conexión. Si hay un problema de ingestión con la salida, Logstash o Elasticsearch, Filebeat ralentizará la lectura de los ficheros.

¿Cómo garantiza Filebeat al menos una entrega?

Filebeat garantiza que los eventos se entregarán a la salida configurada al menos una vez y sin pérdida de datos. Filebeat puede lograr este comportamiento porque almacena el estado de entrega de cada evento en el fichero de registro.

En situaciones donde la salida definida está bloqueada y no ha confirmado todos los eventos, Filebeat seguirá intentando enviar eventos hasta que la salida reconozca que ha recibido los eventos.

Si Filebeat se apaga mientras está enviando eventos, no espera a que la salida reconozca todos los eventos antes de apagarse. Todos los eventos que se envían a la salida, pero que no se reconocen antes de que Filebeat se apague, se envían nuevamente cuando se reinicia Filebeat. Esto garantiza que cada evento se envíe al menos una vez, pero puede terminar con eventos duplicados que se envían a la salida. Se puede configurar Filebeat para que espere un tiempo específico antes de apagarse, configurando la opción “shutdown_timeout”.

Hay una limitación a la garantía de “at-least-once delivery” entrega al menos una vez de Filebeat que involucra la rotación de registros y la eliminación de ficheros antiguos. Si los ficheros de registro se escriben en el disco y se rotan más rápido de lo que Filebeat puede procesar, o si los ficheros se eliminan mientras la salida no está disponible, los datos podrían perderse.

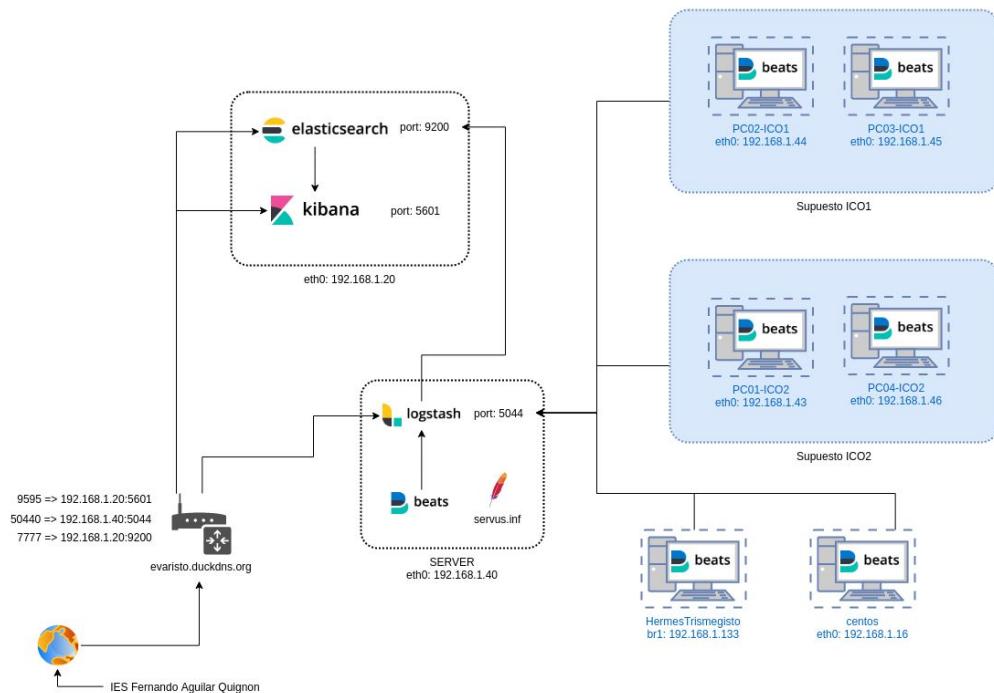
3. Fase de pruebas

En la última semana de mayo se procedió al despliegue de Beats en todos los equipos del departamento, y estuvieron enviando datos continuamente al Logstash y el Elasticsearch que estaban corriendo en el "entorno de tests" durante las semanas consecutivas.

Para ello se preparó un procedimiento para que el administrador del departamento desplegara los agentes Beats.

El resultado fue muy satisfactorio, ya que no se observó ninguna degradación del estado general de la red del departamento de informática, ni en la red externa del "entorno de tests".

Observamos a continuación el esquema de red ya mostrado anteriormente, para refrescar la configuración de red:



4. Documentación del sistema

4.1. Introducción a la aplicación.

Ya conocemos tanto el origen²⁸ como el funcionamiento interno de los componentes principales de ELK Stack. A continuación procederemos con la instalación. Para que no tengamos problemas seguiremos la siguiente recomendación:

“Elasticsearch recomienda que la memoria asignada al proceso de Elasticsearch esté cerca de la mitad de la memoria total disponible.”

Lo más recomendable es ir a:

<https://opendistro.github.io/for-elasticsearch/downloads.html> donde obtendremos las guías actualizadas para las últimas versiones. A continuación veremos el procedimiento a seguir de base.

4.2 Manual de instalación.

Ante cualquier duda con referencia a los ficheros de configuración consultar el repositorio²⁹ del presente proyecto.

Elasticsearch

Elasticsearch (solo código con licencia Apache 2.0) - 6.7.1

Incluye seguridad, alertas, SQL y analizador de rendimiento

RPM (Centos)

```
cd /etc/yum.repos.d/  
sudo curl https://d3g5vo6xdbdb9a.cloudfront.net/yum/opendistroforelasticsearch-artifacts.repo -o opendistroforelasticsearch-artifacts.repo
```

Open Distro para Elasticsearch requiere el Kit completo de desarrollo de Java (JDK), no solo el Java Runtime Environment (JRE). Si no tenemos instalado el JDK, instalaremos la versión 8 o la versión 11:

Java 11

```
sudo yum install java-11-openjdk-devel
```

Java 8

```
sudo yum install java-1.8.0-openjdk-devel
```

²⁸ Para profundizar en la historia completa de ELK Stack ver: Anexo 2

²⁹ Repositorio de éste proyecto: https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro

Si instalamos Java 8 , ejecutaremos el siguiente comando:

```
sudo ln -s /usr/lib/jvm/java-1.8.0/lib/tools.jar /usr/share/elasticsearch/lib/
```

Para iniciar Open Distro para Elasticsearch:

```
sudo systemctl start elasticsearch.service
```

Envíamos solicitudes al servidor para verificar que Elasticsearch esté en funcionamiento:

```
curl -XGET https://localhost:9200 -u admin:admin --insecure
[root@opendistroELK evaristo]# curl -XGET https://localhost:9200 -u admin:admin --insecure
{
  "name" : "uUzpDuY",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Rpf0TvsyQ_Ss_MJcdv1J9g",
  "version" : {
    "number" : "6.7.1",
    "build_flavor" : "oss",
    "build_type" : "rpm",
    "build_hash" : "2f32220",
    "build_date" : "2019-04-02T15:59:27.961366Z",
    "build_snapshot" : false,
    "lucene_version" : "7.7.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}

curl -XGET https://localhost:9200/_cat/nodes?v -u admin:admin --insecure
[root@opendistroELK evaristo]# curl -XGET https://localhost:9200/_cat/nodes?v -u admin:admin --insecure
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
192.168.1.20      46       95     33    1.03    1.16    1.04 mdi      *      uUzpDuY
[root@opendistroELK evaristo]# 

curl -XGET https://localhost:9200/_cat/plugins?v -u admin:admin --insecure
[root@opendistroELK evaristo]# curl -XGET https://localhost:9200/_cat/plugins?v -u admin:admin --insecure
name   component           version
uUzpDuY opendistro_alerting 0.9.0.0
uUzpDuY opendistro_performance_analyzer 0.9.0.0
uUzpDuY opendistro_security 0.9.0.0
uUzpDuY opendistro_sql     0.9.0.0
[root@opendistroELK evaristo]# 
```

Para comprobar el estado del servicio:

```
systemctl status elasticsearch.service
```

Es posible que observemos algunos errores si utilizamos Java 8. Si el servicio sigue siendo fijo active (running), podemos ignorarlos de manera segura:

```
elasticsearch[3969]: java.security.policy: error adding Entry:
elasticsearch[3969]: java.net.MalformedURLException: unknown protocol: jrt
elasticsearch[3969]: java.security.policy: error adding Entry:
elasticsearch[3969]: java.net.MalformedURLException: unknown protocol: jrt
```

Para detener Open Distro for Elasticsearch:

```
sudo systemctl stop elasticsearch.service
```

Para ejecutar Open Distro para Elasticsearch cuando se inicie el sistema:

```
sudo /bin/systemctl daemon-reload
```

```
sudo /bin/systemctl enable elasticsearch.service
```

¿Dónde están los ficheros?

El paquete RPM instala ficheros en las siguientes ubicaciones:

- Inicio de Elasticsearch, scripts de gestión y complementos
 - /usr/share/elasticsearch/
- Ficheros de configuración
 - /etc/elasticsearch
- Variables de entorno
 - /etc/sysconfig/elasticsearch
- Logs
 - /var/log/elasticsearch
- Datos de fragmentos
 - /var/lib/elasticsearch

Ubuntu/Debian

Estos pasos asumen que estamos usando Ubuntu 18.04.

Instalar Java 11:

```
sudo add-apt-repository ppa:openjdk-r/ppa  
sudo apt update  
sudo apt install openjdk-11-jdk
```

Descargar y agregar claves de firma para los repositorios:

```
wget -qO - https://d3g5vo6xdbdb9a.cloudfront.net/GPG-KEY-opendistroforelasticsearch | sudo apt-key add -
```

Añadir los repositorios:

```
echo "deb https://d3g5vo6xdbdb9a.cloudfront.net/apt stable main" | sudo tee -a  
/etc/apt/sources.list.d/opendistroforelasticsearch.list
```

Instalar Elasticsearch OSS:

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-oss-6.7.1.deb  
sudo dpkg -i elasticsearch-oss-6.7.1.deb
```

Instalar Open Distro para Elasticsearch:

```
sudo apt-get update  
sudo apt install opendistroforelasticsearch
```

Para iniciar Open Distro para Elasticsearch:

```
sudo systemctl start elasticsearch.service
```

Enviar solicitudes al servidor para verificar que Elasticsearch esté en funcionamiento:

```
curl -XGET https://localhost:9200 -u admin:admin --insecure
[root@opendistroELK evaristo]# curl -XGET https://localhost:9200 -u admin:admin --insecure
{
  "name" : "uUzpDuY",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Rpf0TvsyQ_Ss_MJcdv1J9g",
  "version" : {
    "number" : "6.7.1",
    "build_flavor" : "oss",
    "build_type" : "rpm",
    "build_hash" : "2f32220",
    "build_date" : "2019-04-02T15:59:27.961366Z",
    "build_snapshot" : false,
    "lucene_version" : "7.7.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}

curl -XGET https://localhost:9200/_cat/nodes?v -u admin:admin --insecure
[root@opendistroELK evaristo]# curl -XGET https://localhost:9200/_cat/nodes?v -u admin:admin --insecure
ip           heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
192.168.1.20      46      95     33     1.03     1.16     1.04 mdi      *      uUzpDuY
[root@opendistroELK evaristo]# ■

curl -XGET https://localhost:9200/_cat/plugins?v -u admin:admin --insecure
[root@opendistroELK evaristo]# curl -XGET https://localhost:9200/_cat/plugins?v -u admin:admin --insecure
name  component          version
uUzpDuY opendistro_alerting 0.9.0.0
uUzpDuY opendistro_performance_analyzer 0.9.0.0
uUzpDuY opendistro_security 0.9.0.0
uUzpDuY opendistro_sql      0.9.0.0
[root@opendistroELK evaristo]# ■
```

Para comprobar el estado del servicio:

```
systemctl status elasticsearch.service
```

Para detener Open Distro for Elasticsearch:

```
sudo systemctl stop elasticsearch.service
```

Para ejecutar Open Distro para Elasticsearch cuando se inicie el sistema:

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
```

El paquete Debian instala ficheros en las siguientes ubicaciones:

- Inicio de Elasticsearch, scripts de gestión y complementos
 - /usr/share/elasticsearch/
- Ficheros de configuración
 - /etc/elasticsearch
- Variables de entorno
 - /etc/default/elasticsearch
- Logs
 - /var/log/elasticsearch
- Datos de fragmentos
 - /var/lib/elasticsearch

Notas sobre Debian

Si utilizamos Debian en lugar de Ubuntu, es probable que necesitemos hacer algunas modificaciones en el proceso de instalación.

Al instalar Java 11, en lugar de:

```
sudo add-apt-repository ppa:openjdk-r/ppa
```

ejecutaríamos:

```
sudo echo 'deb http://deb.debian.org/debian stretch-backports main' > /etc/apt/sources.list.d/backports.list
```

Antes de instalar Open Distro para Elasticsearch, ejecutaríamos:

```
apt install apt-transport-https
```

Ejemplo de configuración:

Tras haber generado nuestros propios certificados, siguiendo el procedimiento³⁰ que nos proporciona Opendistro, ésta es la configuración:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/elasticsearch/elasticsearch.yml

Éstos certificados los deberemos de colocar en todos los sistemas en los que vayan a correr cualquiera de los procesos que se comunicarán con Elasticsearch (Logstash, Metricbeat, etc...)

El montón de memoria JVM fue configurado para usar 4GB en éste ejemplo en el que la MV contaba con 8 GB:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/elasticsearch/jvm.options

Kibana

RPM (Centos) y Ubuntu/Debian

Kibana (solo código con licencia Apache 2.0) - 6.7.1

Incluye seguridad y alertas

Lo más recomendable es ir a:

<https://opendistro.github.io/for-elasticsearch/downloads.html> donde obtendremos las guías actualizadas para las últimas versiones.

³⁰ Crear certificados propios:

<https://aws.amazon.com/es/blogsopensource/add-ssl-certificates-open-distro-for-elasticsearch/>

Si aún no lo hemos hecho, agregaremos los repositorios yum especificados anteriormente.

```
sudo yum install opendistroforelasticsearch-kibana
```

O

```
sudo apt install opendistroforelasticsearch-kibana
```

(Opcional) Modificar /etc/kibana/kibana.yml.

```
sudo systemctl start kibana.service
```

Para detener Kibana:

```
sudo systemctl stop kibana.service
```

Para ejecutar Kibana cuando se inicia el sistema:

```
sudo /bin/systemctl daemon-reload
```

```
sudo /bin/systemctl enable kibana.service
```

Podemos modificar los valores en /etc/kibana/kibana.yml.

Después de iniciar Kibana, podemos acceder a él en el puerto 5601. Por ejemplo, <http://localhost:5601>

Iniciamos sesión con el nombre de usuario admin y la contraseña predeterminada admin.

Podemos seleccionar “Probar datos de muestra” o cerrar esta ventana de demostración para comenzar a recibir nuestros propios datos.

The screenshot shows the Kibana interface with the title "Add Data to Kibana". At the top, there is a navigation bar with tabs: Home, Add data, All, Logging, Metrics, Security analytics, and Sample data (which is currently selected). Below the navigation bar, there are three cards, each representing a different type of sample data:

- Sample eCommerce orders:** Describes sample data for tracking eCommerce orders. It includes a small dashboard preview and an "Add" button.
- Sample flight data:** Describes sample data for monitoring flight routes. It includes a small dashboard preview and an "Add" button.
- Sample web logs:** Describes sample data for monitoring web logs. It includes a small dashboard preview and an "Add" button.

Ejemplo de configuración de Kibana:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/kibana/kibana.yml

Beats

Planteamiento previo del contexto encontrado

Todos los “Beats” comparten un procedimiento similar de configuración, variando únicamente en los módulos internos de cada Beat que se puedan activar (Opcionalmente)

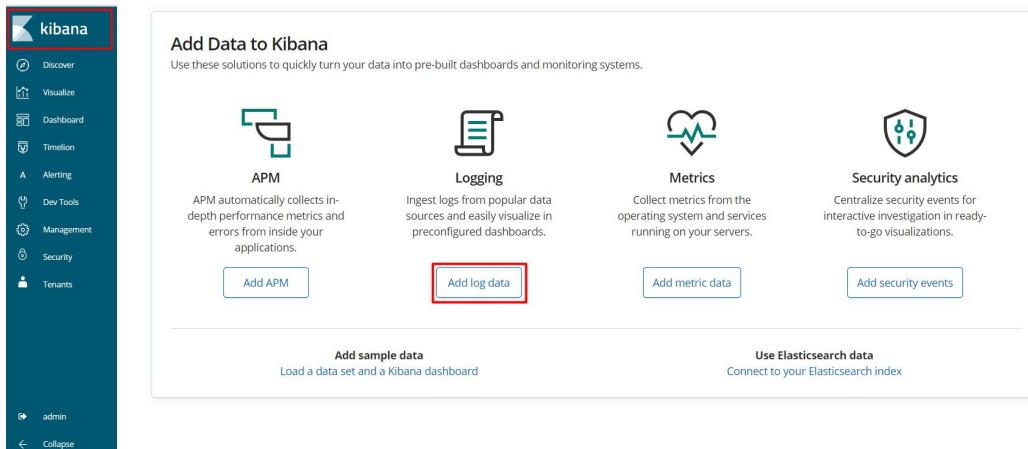
En los siguientes pasos veremos como implementar Filebeat, sirviendo este ejemplo para la implantación de cualquier otro “Beat” en GNU-Linux.

En la página principal de Kibana observamos accesos directos de cortesía para los procedimientos de instalación de:

- APM
 - Rendimiento de aplicaciones.
- Loggin
 - Filebeat y sus módulos.
- Metrics
 - Metricbeat y sus módulos.
- Security analytics
 - Módulos específicos de Filebeat para herramientas de seguridad.

Dichos procedimientos tienen una clara función, dar un aspecto más intuitivo a Kibana, de cara al usuario final, pero no son 100% funcionales, y menos en nuestra versión “Opendistro”, por lo que vamos a analizar hasta qué punto nos pueden ser útiles.

Cada uno de los accesos directos, que observamos en la siguiente imagen, nos llevarán a las distintas guías de instalación por ejemplo si queremos monitorizar logs del sistema, intuitivamente seleccionaremos:



A continuación, en función de nuestro sistema, elegiríamos si trabajamos con paquetes “macOS”, “DEB” o “RPM”

Realmente la única diferencia entre elegir un módulo u otro, radica en el paso 3, donde nos indica “Activar el módulo” Observamos a continuación la diferencia entre el módulo “system” y el módulo “postgresql”

3 Enable and configure the system module

From the installation directory, run:

```
./filebeat modules enable system
```

[Copy snippet](#)

Modify the settings in the `modules.d/system.yml` file.

3 Enable and configure the postgresql module

From the installation directory, run:

```
./filebeat modules enable postgresql
```

[Copy snippet](#)

Modify the settings in the `modules.d/postgresql.yml` file.

Pero en cualquier módulo de la sección “Loggin” y “Security analitics” es Flebeat lo que estaremos configurando, o Metricbeat en cualquier módulo de la sección “Métrics”.

Observando el procedimiento, encontraremos un primer problema en el punto 1. Pues recientemente, en Elastic, agregaron una verificación de licencia que hace que los agentes de beats de edición predeterminada no sean compatibles con la versión OSS (OSSFL: Open Source Software for Life) de Elasticsearch que tenemos instalada al haber elegido “Opendistro”.

En la siguiente imagen se aprecia que el enlace proporcionado no nos conviene:

The `system` Metricbeat module collects CPU, memory, network, and disk statistics from the host. It collects system wide statistics and statistics per process and filesystem. Learn more.

[View exported fields](#)

Self managed Elastic Cloud

Getting Started

macOS DEB RPM Windows

1 Download and install Metricbeat

First time using Metricbeat? See the [Getting Started Guide](#). <https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-oss-6.7.1-amd64.deb> [Copy snippet](#)

```
curl -L -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.7.1-amd64.deb
sudo dpkg -i metricbeat-6.7.1-amd64.deb
```

Looking for the 32-bit packages? See the [Download page](#).

Es probable que en futuras versiones de “Opendistro for Elasticsearch” editen esta sección, pero por ahora debemos de asegurarnos de que nos descargamos la versión correcta OSS (OSSFL: Open Source Software for Life) de nuestro Beat.

En caso de descargar una versión predeterminada de un Beat, en vez de la versión OSS, nos encontraríamos con el siguiente error al levantar el pertinente servicio Beat “failed: cannot retrieve the elasticsearch license”

```
2019-05-23T13:43:19.429+0200  INFO  elasticsearch/client.go:739  Attempting to connect to Elasticsearch version 6.7.1
2019-05-23T13:43:48.835+0200  INFO  [monitoring]  log/log.go:144  Non-zero metrics in the last 30s
{"monitoring": {"metrics": {"beat": {"cpu": {"system": {"ticks": 210, "time": {"ms": 196}}, "total": {"ticks": 480, "time": {"ms": 220}}, "value": 480}, "user": {"ticks": 270, "time": {"ms": 124}}, "handles": {"limit": {"hard": 4996, "soft": 1024}, "open": 7}, "info": {"ephemeral_id": "46c294d5-f23e-43d0-9e94-d05964af2fb9"}, "uptime": {"ms": 60054}}, "memstats": {"gc_next": 9980992, "memory_alloc": 4992928, "memory_total": 67041808, "rss": 528384}, "libbeat": {"config": {"module": {"running": 0}}, "output": {"read": {"bytes": 1166}, "write": {"bytes": 344}}, "pipeline": {"clients": 6, "events": 47, "active": 100, "published": 47, "total": 47}}, "metricbeat": {"system": {"cpu": {"events": 3, "success": 3}, "load": {"events": 3, "success": 3}, "memory": {"events": 3, "success": 3}, "network": {"events": 16, "success": 6}, "process": {"events": 26, "success": 26}, "process_summary": {"events": 3, "success": 3}, "socket_summary": {"events": 3, "success": 3}}, "system": {"load": {"1": 0.09, "15": 0.01, "5": 0.05}, "norm": {"1": 0.09, "15": 0.01, "5": 0.05}}}}, "2019-05-23T13:43:49.073+0200  ERROR  pipeline/outnut.go:108  Failed to connect to backoff(elasticsearch(https://192.168.1.20:9200)): Connection marked as failed because the onConnect callback failed: cannot retrieve the elasticsearch license, error from server, response code: 500
2019-05-23T13:43:49.073+0200  INFO  pipeline/output.go:93  Attempting to reconnect to backoff(elasticsearch(https://192.168.1.20:9200)) with 5 reconnect attempts(s)
^C
```

Se solucionaría instalando la versión OSS del mismo Beat.

Planteado esto, ahora sí, veremos el procedimiento para la implantación de cualquier Beat.

Descarga del Beat (Filebeat)

Encontramos el catálogo completo de descarga en:

<https://www.elastic.co/es/downloads/past-releases>

Filtraremos por el Beat que necesitemos (**OSS**), así como la versión que coincida con la versión de nuestro Elasticsearch.

Past Releases

Filebeat OSS 6.7.1

April 04, 2019

▶ See Release Notes Download

Filebeat OSS 6.7.1

DEB 32-BIT sha

DEB 64-bit Abrir enlace en una pestaña nueva

RPM 32-bit

RPM 64-bit

LINUX 32-bit

LINUX 64-bit

MAC OS X

WINDOWS 64-BIT sha

Copiar dirección de enlace

Copiaremos el enlace y lo descargaremos e instalaremos:

```
sudo curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-oss-6.7.1-amd64.deb  
sudo dpkg -i filebeat-oss-6.7.1-amd64.deb
```

Editando la configuración

Modificamos /etc/filebeat/filebeat.yml

Definiendo Inputs

```
#===== Filebeat inputs =====
filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

- type: log
  # Change to true to enable this input configuration.
  enabled: false
  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*
```

Añadiremos las rutas de los ficheros o directorios con ficheros de log que queremos monitorizar (en caso de añadir alguna línea, no debemos de olvidarnos de cambiar el estado de “enabled” a “true”).

Por ejemplo, en una máquina con un “squid” se configuró de la siguiente manera:

```
#===== Filebeat inputs =====
filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

- type: log
  # Change to true to enable this input configuration.
  enabled: true
  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/squid/access.log
    #- /var/log/auth.log
    #- c:\programdata\elasticsearch\logs\*
```

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/filebeat/filebeat.yml

Hay ciertos ficheros de logs que estarán preasignados por los “módulos” de Filebeat.

Un ejemplo de esto sería el módulo “system” que por lo que hemos podido comprobar, recolecta datos de:

- Debian/Ubuntu
 - /var/log/auth (Autenticaciones) y /var/log/syslog (Mensajes del sistema)
- RedHat/CentOS
 - /var/log/auth (Autenticaciones) y /var/log/message (Mensajes del sistema)

Podría darse el caso de especificar en estas líneas: - /var/log/syslog y además activar el módulo system, lo que resultaría en un duplicado de información.

En /etc/filebeat/modules.d/ podremos observar los módulos disponibles:

```
evaristo@PC01-IC02:~$ ls -l /etc/filebeat/modules.d/
total 76
-rw-r--r-- 1 root root 371 abr 2 17:00 apache2.yml.disabled
-rw-r--r-- 1 root root 175 abr 2 17:00 auditd.yml.disabled
-rw-r--r-- 1 root root 1250 abr 2 17:00 elasticsearch.yml.disabled
-rw-r--r-- 1 root root 269 abr 2 17:00 haproxy.yml.disabled
-rw-r--r-- 1 root root 546 abr 2 17:00 icinga.yml.disabled
-rw-r--r-- 1 root root 371 abr 2 17:00 iis.yml.disabled
-rw-r--r-- 1 root root 257 abr 2 17:00 iptables.yml.disabled
-rw-r--r-- 1 root root 396 abr 2 17:00 kafka.yml.disabled
-rw-r--r-- 1 root root 188 abr 2 17:00 kibana.yml.disabled
-rw-r--r-- 1 root root 563 abr 2 17:00 logstash.yml.disabled
-rw-r--r-- 1 root root 189 abr 2 17:00 mongodb.yml.disabled
-rw-r--r-- 1 root root 368 abr 2 17:00 mysql.yml.disabled
-rw-r--r-- 1 root root 569 abr 2 17:00 nginx.yml.disabled
-rw-r--r-- 1 root root 388 abr 2 17:00 osquery.yml.disabled
-rw-r--r-- 1 root root 192 abr 2 17:00 postgresql.yml.disabled
-rw-r--r-- 1 root root 463 abr 2 17:00 redis.yml.disabled
-rw-r--r-- 1 root root 190 abr 2 17:00 suricata.yml.disabled
-rw-r--r-- 1 root root 574 abr 2 17:00 system.yml
-rw-r--r-- 1 root root 195 abr 2 17:00 traefik.yml.disabled
evaristo@PC01-IC02:~$
```

Podemos personalizar el módulo, definiendo nuevas rutas.

```
- module: system
# Syslog
syslog:
  enabled: true

  # Set custom paths for the log files. If left empty,
  # Filebeat will choose the paths depending on your OS.
  #var.paths:

  # Convert the timestamp to UTC. Requires Elasticsearch >= 6.1.
  #var.convert_timezone: false

# Authorization logs
auth:
  enabled: true

  # Set custom paths for the log files. If left empty,
  # Filebeat will choose the paths depending on your OS.
  #var.paths:

  # Convert the timestamp to UTC. Requires Elasticsearch >= 6.1.
  #var.convert_timezone: false
/etc/filebeat/modules.d/system.yml (END)
```

En el presente ejemplo hemos dejado la sección “Filebeat Input” de /etc/filebeat/filebeat.yml por defecto (enabled: false) ya que activaremos el módulo “system” más adelante.

Definiendo Outputs

Debemos elegir a dónde apuntar la salida (Logstash o Elasticsearch) No podemos dejar las dos configuraciones sin comentar, una de las dos deberá de estar comentada³¹ o veremos un error.

Por defecto nos viene descomentado Elasticsearch y comentado Logstash.

```
#----- Outputs -----
# Configure what output to use when sending the data collected by the beat.

#----- Elasticsearch output -----
#output.elasticsearch:
#  # Array of hosts to connect to.
#  #hosts: ["localhost:9200"]

#  # Enabled ilm (beta) to use index lifecycle management instead daily indices.
#  #ilm.enabled: false

#  # Optional protocol and basic auth credentials.
#  #protocol: "https"
#  #username: "elastic"
#  #password: "changeme"

#----- Logstash output -----
output.logstash:
#  # The Logstash hosts
hosts: ["192.168.1.40:5044"]

#  # Optional SSL. By default is off.
#  # List of root certificates for HTTPS server verifications
ssl.certificateAuthorities: ["/home/evaristo/MyRootCA.pem"]

#  # Certificate for SSL client authentication
ssl.certificate: "/home/evaristo/odfe-node1.pem"

#  # Client Certificate Key
ssl.key: "/home/evaristo/odfe-node1.key"

sslverification_mode: "none"
```

Config para apuntar a Elasticsearch

```
#----- Elasticsearch output -----
output.elasticsearch:

#  # Array of hosts to connect to.
hosts: ["IP_ESCALICSEARCH:9200"]

#  # Enabled ilm (beta) to use index lifecycle management instead daily indices.
#  #ilm.enabled: false

#  # Optional protocol and basic auth credentials.
#  protocol: "https"
#  username: "admin"
#  password: "admin"

#  # Optional SSL. By default is off.
```

³¹ Un fallo común es comentar(descomentar) todas las líneas menos la primera, la del nombre de la salida (output.elasticsearch o output.logstash), que debemos de comentar/descomentar también.

```
# List of root certificates for HTTPS server verifications
ssl.certificateAuthorities: ["RUTA_A_/MyRootCA.pem"]

# Certificate for SSL client authentication
ssl.certificate: "RUTA_A_/odfe-node1.pem"

# Client Certificate Key
ssl.key: "RUTA_A_/odfe-node1.key"

ssl.verificationMode: "none"
```

Config para apuntar a Logstash

```
#----- Logstash output -----
output.logstash:
  # The Logstash hosts
  hosts: ["IP_LOGSTASH:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  ssl.certificateAuthorities: ["RUTA_A_/MyRootCA.pem"]

  # Certificate for SSL client authentication
  ssl.certificate: "RUTA_A_/odfe-node1.pem"

  # Client Certificate Key
  ssl.key: "RUTA_A_/odfe-node1.key"

  ssl.verificationMode: "none"
```

Habilitar un módulo predeterminado (opcional)

El paso 3, es donde se activa el módulo.

En nuestro caso (DEB) sería:

`sudo filebeat modules enable system`

Como bien nos sugiere el procedimiento integrado en Kibana:

- 3 Enable and configure the system module

[Copy snippet](#)

```
sudo filebeat modules enable system
```

Modify the settings in the `/etc/filebeat/modules.d/system.yml` file.

```
evaristo@PC01-IC02:~$ sudo filebeat modules enable system
Enabled system
evaristo@PC01-IC02:~$
```

Ejecutar “filebeat modules enable system” es similar a renombrar `/etc/filebeat/modules.d/system.yml.disabled` por `/etc/filebeat/modules.d/system.yml`

Instalando paneles en Kibana (Opcional)

Si deseamos instalar los paneles por defecto, deberemos de tener activada la “Salida para Elasticsearch” en /etc/filebeat/filebeat.yml y además declarar la dirección o nombre DNS del servidor Kibana:

```
#===== Dashboards =====
# These settings control loading the sample dashboards to the Kibana index. Loading
# the dashboards is disabled by default and can be enabled either by setting the
# options here, or by using the '-setup' CLI flag or the 'setup' command.
#setup.dashboards.enabled: false

# The URL from where to download the dashboards archive. By default this URL
# has a value which is computed based on the Beat name and version. For released
# versions, this URL points to the dashboard archive on the artifacts.elastic.co
# website.
#setup.dashboards.url:

#===== Kibana =====

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify an additional path, the scheme is required: http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "192.168.1.20:5601"

  # Kibana Space ID
  # ID of the Kibana Space into which the dashboards should be loaded. By default,
  # the Default Space will be used.
  #space.id:
```

Como observamos en el “Paso 4” del procedimiento en Kibana, nos sugiere ejecutar:

sudo filebeat setup

4 Start Filebeat

The `setup` command loads the Kibana dashboards. If the dashboards are already set up, omit this command.

[Copy snippet](#)

```
sudo filebeat setup
sudo service filebeat start
```

Pero de hacerlo obtendremos el error “Error checking if xpand is available: 500 Internal Server Error”

```
root@SERVER:/home/evaristo# sudo filebeat setup
Loaded index template
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Exiting: 4 errors: [Error checking if xpand is available: 500 Internal Server Error] {"error": {"root_cause": [{"type": "security_exception", "reason": "Unexpected exception indices:admin/get"}]}, "type": "security_exception", "reason": "Unexpected exception indices:admin/get"}, Error checking if xpand is available: 500 Internal Server Error: {"error": {"root_cause": [{"type": "security_exception", "reason": "Unexpected exception indices:admin/get"}]}, "type": "security_exception", "reason": "Unexpected exception indices:admin/get"}, Error checking if xpand is available: 500 Internal Server Error: {"error": {"root_cause": [{"type": "security_exception", "reason": "Unexpected exception indices:admin/get"}]}, "type": "security_exception", "reason": "Unexpected exception indices:admin/get"}, Error checking if xpand is available: 500 Internal Server Error: {"error": {"root_cause": [{"type": "security_exception", "reason": "Unexpected exception indices:admin/get"}]}, "type": "security_exception", "reason": "Unexpected exception indices:admin/get"}]
```

En su lugar ejecutaremos:

sudo filebeat setup -e --dashboards --pipelines --template

Una cómoda práctica, para la primera instalación, sería configurar al menos una vez, la salida a Elasticsearch, con la intención de dejar instalar los paneles predeterminados de Kibana configurados, una vez hecho esto ya cambiaríamos la salida a Logstash, en el caso de que necesitáramos hacer un tratamiento de datos (como suele ser bastante común con Filebeat, y no tan común con Metricbeat)

Arrancando Filebeat

```
sudo service filebeat start
```

O

```
sudo systemctl start filebeat
```

```
evaristo@PC01-IC02:~$ sudo systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
     Active: active (running) since mar 2019-05-21 16:13:05 CEST; 23h ago
       Docs: https://www.elastic.co/products/beats/filebeat
   Main PID: 1031 (filebeat)
     CGroup: /system.slice/filebeat.service
             └─1031 /usr/share/filebeat/bin/filebeat -c /etc/filebeat/filebeat.yml -path.home /usr/share/filebeat -path.config /etc/filebeat -path.data /var/lib/fil
may 21 16:13:05 PC01-IC02 systemd[1]: Started Filebeat sends log files to Logstash or directly to Elasticsearch..
Lines 1-9/9 (END)
```

Si queremos que Filebeat se ejecute al inicio:

```
sudo systemctl enable filebeat
```

```
evaristo@PC01-IC02:~$ sudo systemctl enable filebeat
Synchronizing state of filebeat.service with SysV init with /lib/systemd/systemd-sysv-install...
Executing /lib/systemd/systemd-sysv-install enable filebeat
evaristo@PC01-IC02:~$ █
```

Index patterns

En el caso de haber configurado “output.elasticsearch” y de haber ejecutado “sudo filebeat setup -e --dashboards --pipelines --template” tendremos automáticamente en Kibana configurado el index pattern de Filebeat-* así como todo el abanico de gráficas y paneles preconfigurados.

The screenshot shows the Kibana interface with the 'Management' tab selected. On the left, the sidebar has 'Discover', 'Visualize', 'Dashboard', 'Timeline', 'Alerting', 'Dev Tools', 'Management', 'Security', and 'Tenants'. Under 'Management', there are 'admin' and 'Collapse' buttons. The main area shows the 'Index Patterns' section with a 'Create index pattern' button. A list of existing index patterns includes 'authespecial-*', 'auditbeat-*', 'authpersonalitydequeuev0-*', 'filebeat-*' (which is highlighted with a red box), 'journalbeat-*', 'metricbeat-*', 'packetbeat-*', and 'squid-tikitikiv3-*'. To the right, the 'filebeat-*' index pattern details are shown. It lists fields such as '@timestamp', '@version', '_id', '_index', '_score', '_source', '_type', 'apache2.access.agent', 'apache2.access.body_sent.bytes', and 'apache2.access.geoip.city_name'. Each field has its type (date, string, number, etc.), format, searchability, aggregability, and exclusion status indicated by icons. A 'Time Filter field name: @timestamp' section is also present.

En caso contrario, de haber configurado “output.logstash” o simplemente haber prescindido de ejecutar

sudo filebeat setup -e --dashboards --pipelines --template

deberemos proceder de la siguiente manera:

Accedemos a la dirección del servidor Kibana

http://SERVIDOR_KIBANA:9595



El usuario por defecto es:

Usuario: admin

Contraseña: admin

A screenshot of the Kibana 6.7.1 management interface. On the left is a sidebar with various options: Discover, Visualize, Dashboard, Timelion, Alerting, Dev Tools, Management (which is highlighted with a red border), Security, and Tenants. The main content area shows a "Kibana 6.7.1 management" heading with a gear icon, followed by the text "Manage your indices, index patterns, saved objects, Kibana settings, and more." Below this is a note: "A full list of tools can be found in the left menu". Above the main content, under the "Kibana" heading, there are links for "Index Patterns", "Saved Objects", and "Advanced Settings", with "Index Patterns" also having a red border around it.

Accedemos a “Management” y hacemos clic en “Create index pattern”

Escribiremos el nombre del índice en cuestión.

De manera predeterminada, Filebeat está generando un índice nombrados de la siguiente manera: `filebeat-[nº versión]-[YYYY.MM.dd]`. Es decir, en Elasticsearch se está generando un índice por cada día del mes, y éstos son los que está detectando Kibana.

Por lo tanto añadimos un * a la derecha de filebeat- con la intención de que nuestro index pattern detecte todos los índices de filebeat.

Hacemos clic en “Next step”

> Next step

Indicamos cual es el “Time Filter field”

Create index pattern

★ authespecial**
auditbeat-*
authpersonalitydequenuevo*
journalbeat-*
metricbeat-*
packetbeat-*
squid-tikitikiv3
squid-tikitikiv3*

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

Include system indices

Step 2 of 2: Configure settings

You've defined **filebeat*** as your index pattern. Now you can specify some settings before we create it.

Time Filter field name Refresh

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

Hide advanced options

Custom index pattern ID

Kibana will provide a unique identifier for each index pattern. If you do not want to use this unique ID, enter a custom one.

Back **Create index pattern**

En opciones avanzadas se nos da la posibilidad de customizar el ID, de esta forma podríamos recuperar la utilidad de antiguos paneles que tuviéramos configurados para un patrón que en un momento eliminamos, y que desde ese momento quedaron rotos como se aprecia en la siguiente imagen:

kibana

Discover Visualize Dashboard Timelon Alerting Dev Tools Management Security Tenants admin Collapse

Dashboard / [Filebeat System] Syslog dashboard Full screen Share Clone Edit Documentation 5 seconds Last 15 minutes Options Update

Add a filter + Dashboards [Filebeat System] Syslog | Sudo commands | SSH logins | New users and groups

Could not locate that index-pattern (id: filebeat-*), [click here to re-create it](#/management/kibana/index) Could not locate that index-pattern (id: filebeat-*), [click here to re-create it](#/management/kibana/index)

Una vez hecho clic en “Create index pattern” habremos terminado. Opcionalmente podemos establecer éste u otro índice como “índice predeterminado” haciendo clic en el ícono de la estrella señalado en la siguiente imagen:

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	<input type="checkbox"/>
@version	string		●	●	<input type="checkbox"/>
@version.keyword	string		●	●	<input type="checkbox"/>
_id	string		●	●	<input type="checkbox"/>
_index	string		●	●	<input type="checkbox"/>
_score	number				<input type="checkbox"/>
_source	_source				<input type="checkbox"/>
_type	string		●	●	<input type="checkbox"/>
apache2.access.agent	string		●		<input type="checkbox"/>

La afectación que tiene ésto es únicamente para la sección “Discovery” en la que por defecto, nos mostrará el índice predeterminado”

Ejemplos de paneles predefinidos

Veremos a continuación una muestra de los paneles predefinidos que instalamos:

Name	Type
Title	Type
Access logs over time [Filebeat Nginx]	Visual Builder
Access map [Filebeat IIS]	Coordinate Map
Access Map [Filebeat Nginx]	Coordinate Map
Access Map [Filebeat Nginx] [ML]	Coordinate Map
Access Map [Filebeat Traefik]	Coordinate Map
Access Map [Filebeat Traefik] [ML]	Coordinate Map
Backend breakdown [Filebeat HAProxy]	Pie
Browsers breakdown [Filebeat Apache2]	Pie
Browsers breakdown [Filebeat IIS]	Pie
Browsers breakdown [Filebeat Nginx]	Pie

Title	Description	Actions
[Filebeat Kafka] Overview	Filebeat Kafka module dashboard	Edit
Overview [Filebeat MongoDB]	Filebeat MongoDB module overview	Edit
[Filebeat Apache2] Access and error logs	Filebeat Apache2 module dashboard	Edit
[Filebeat Auditd] Audit Events	Dashboard for the Auditd Filebeat module	Edit
[Filebeat System] Sudo commands	Sudo commands dashboard from the Filebeat System module	Edit
[Filebeat PostgreSQL] Overview	Overview dashboard for the Filebeat PostgreSQL module	Edit
[Filebeat PostgreSQL] Query Duration Overview	Dashboard for analyzing the query durations of the Filebeat PostgreSQL module	Edit
[Filebeat Icinga] Debug Log	Filebeat Icinga module dashboard for the debug logs	Edit
[Filebeat Icinga] Main Log	Filebeat Icinga module dashboard for the main log files	Edit
[Filebeat HAProxy] Overview	Filebeat HAProxy module dashboard	Edit
[Filebeat Nginx] Overview	Dashboard for the Filebeat Nginx module	Edit

Paneles Filebeat - módulo system

Los paneles a continuación pertenecen al módulo “system”

Syslog events by hostname [Filebeat System]

Syslog hostnames and processes [Filebeat System]

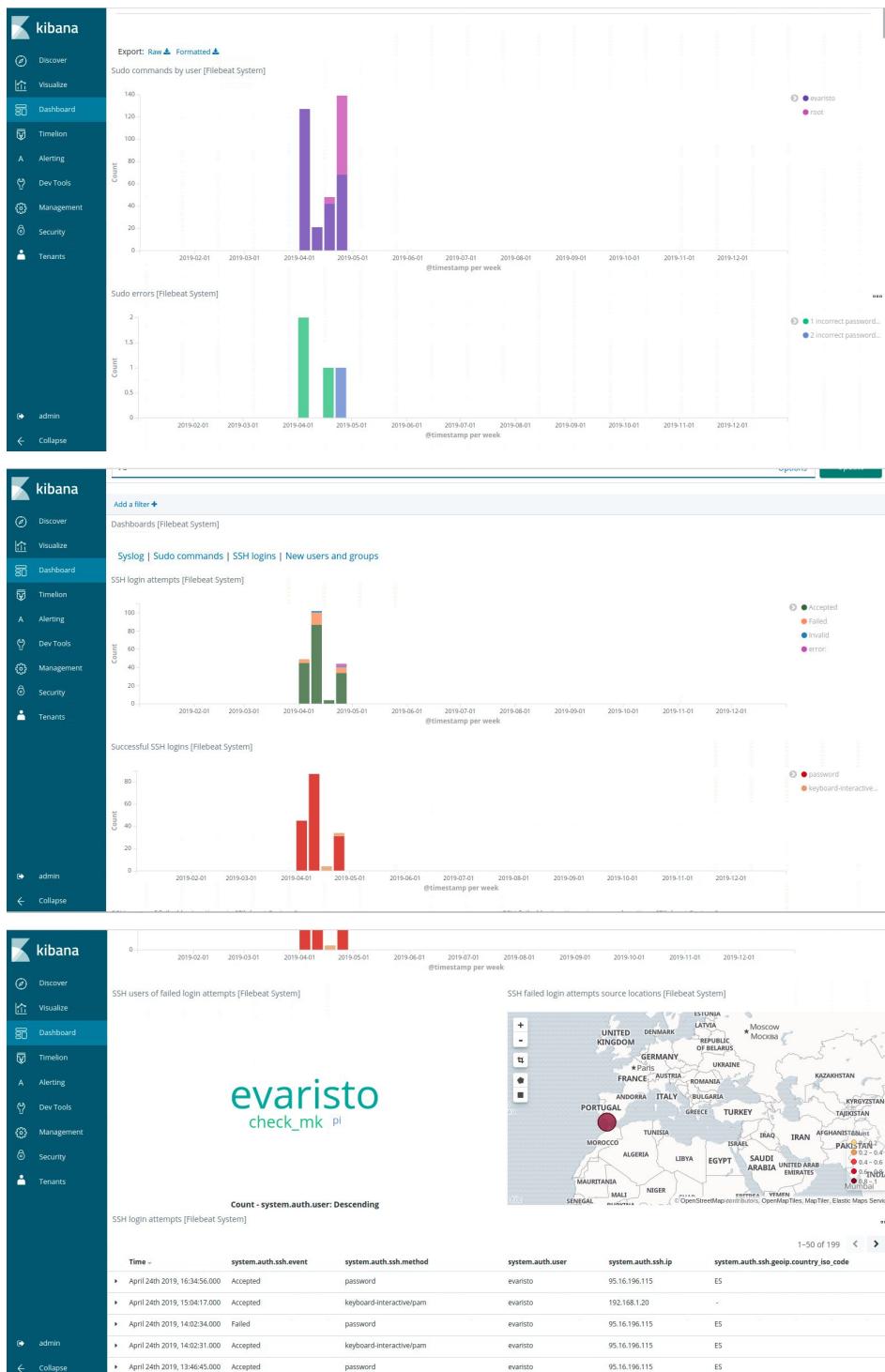
Syslog logs [Filebeat System]

Time	system.syslog.hostname	system.syslog.program	system.syslog.message
April 25th 2019, 08:45:04.000	HermesTrismegisto	wpa_supplicant	wlp40: CTRL-EVENT-SCAN-FAILED ret=-95 retry=1
April 25th 2019, 08:45:03.000	HermesTrismegisto	wpa_supplicant	message repeated 2 times: [wlp40: CTRL-EVENT-SCAN-FAILED ret=-95 retry=1]
April 25th 2019, 08:45:01.000	HermesTrismegisto	wpa_supplicant	wlp40: CTRL-EVENT-SCAN-FAILED ret=-95 retry=1
April 25th 2019, 08:45:01.000	HermesTrismegisto	CRON	(evaristo) CMD (~/_duckdns/ducksh >/dev/null 2>&1)
April 25th 2019, 08:45:00.000	HermesTrismegisto	wpa_supplicant	message repeated 239 times: [wlp40: CTRL-EVENT-SCAN-FAILED ret=-95 retry=1]
April 25th 2019, 08:41:01.000	HermesTrismegisto	wpa_supplicant	wlp40: CTRL-EVENT-SCAN-FAILED ret=-95 retry=1

Dashboard / [Filebeat System] Sudo commands

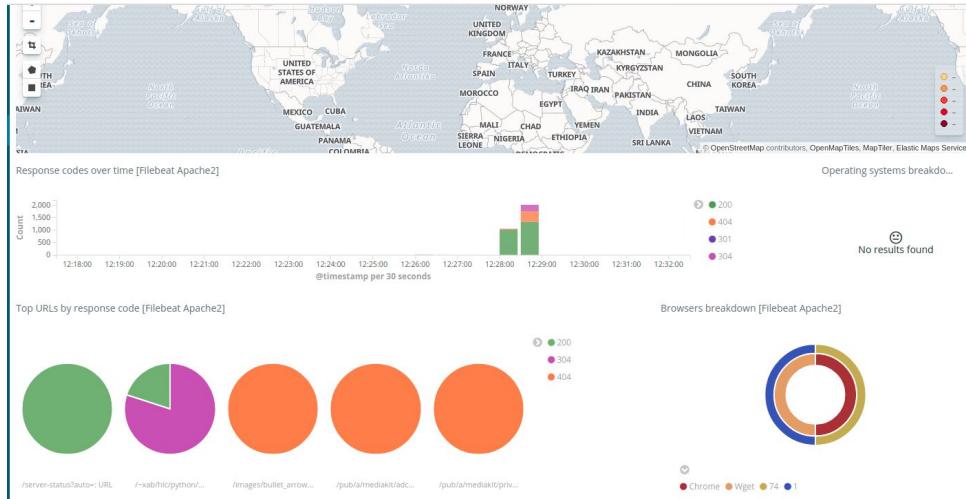
Top sudo commands [Filebeat System]

system.auth.user: Descending	Count
evaristo	41
evaristo	12
evaristo	10
evaristo	10
evaristo	8



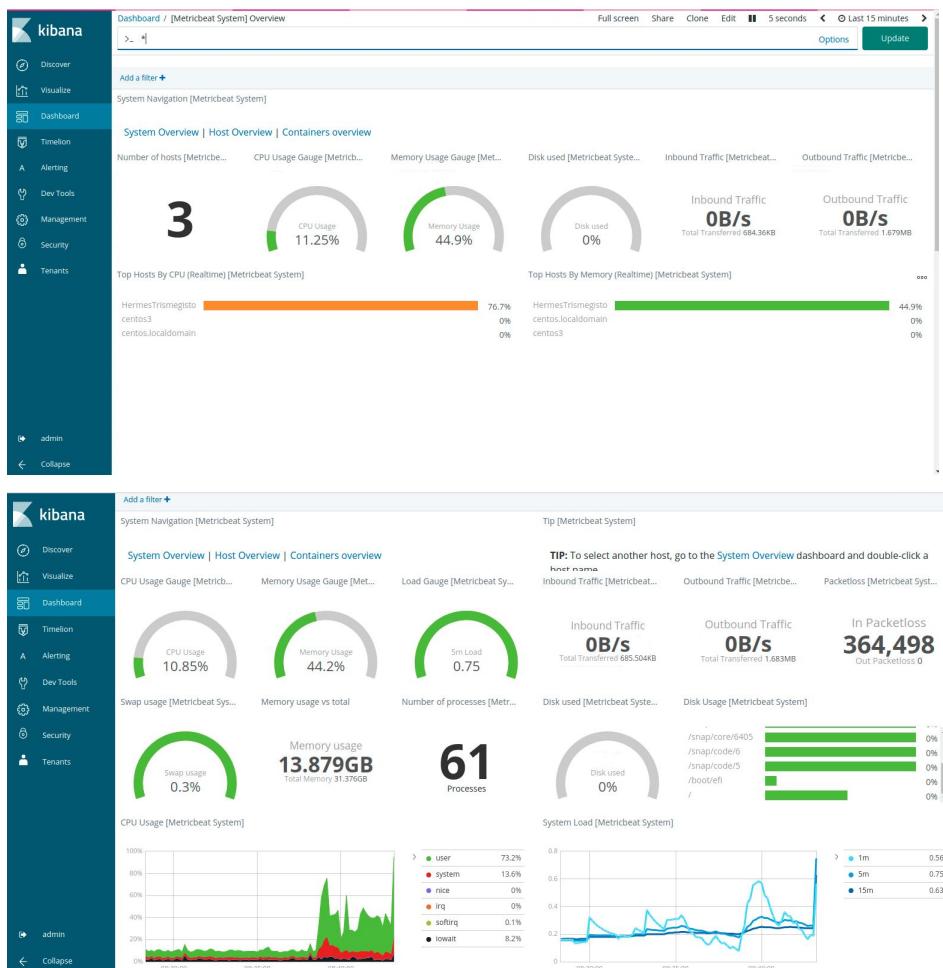
Paneles Filebeat - módulo apache

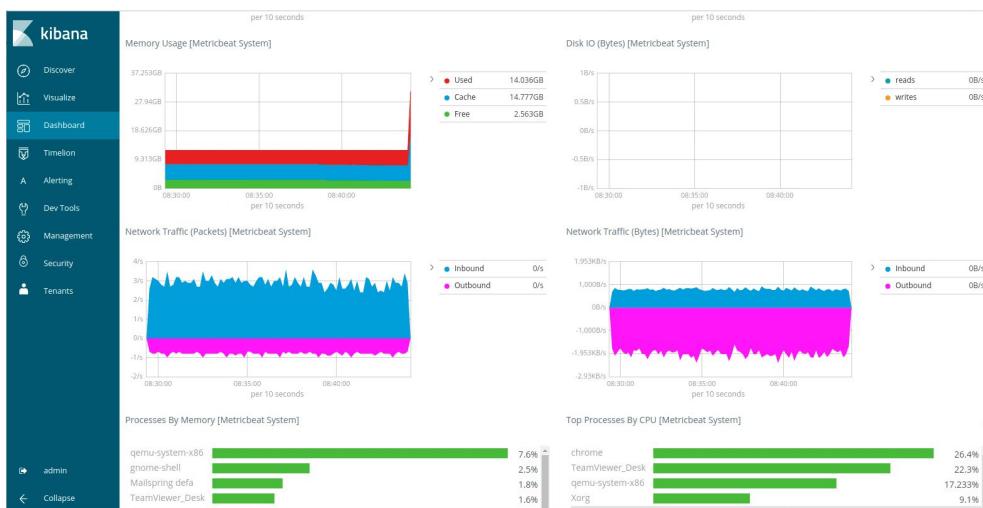
El siguiente panel pertenece al módulo “apache”, de Filebeat.



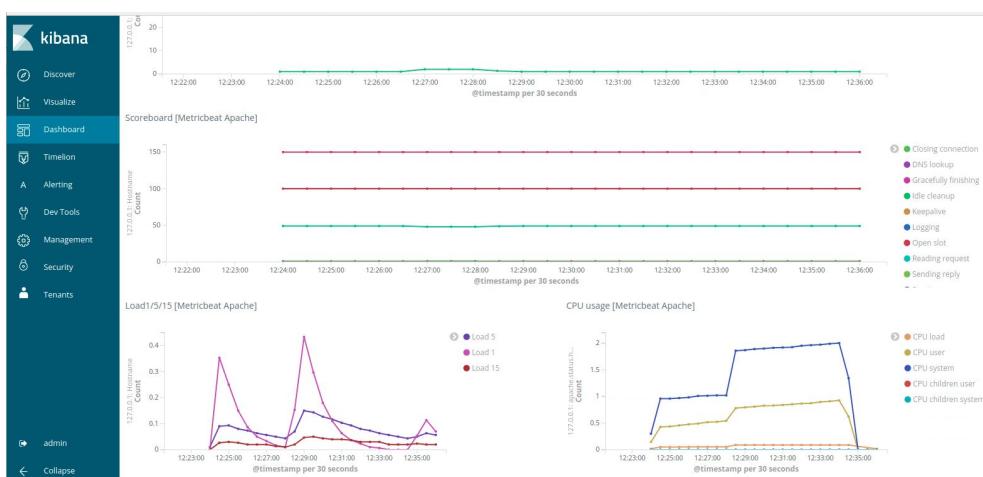
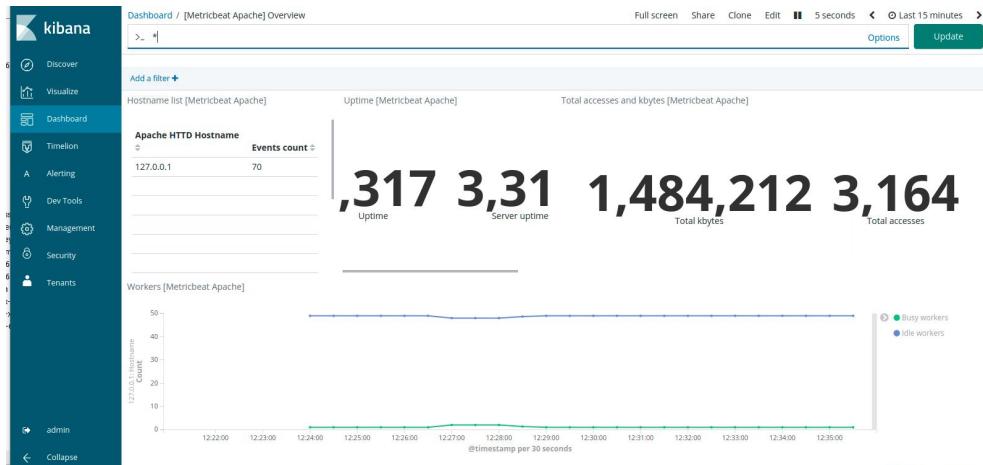
Paneles Metricbeat - módulo system

Los siguientes paneles pertenecen al módulo “system” de Metricbeat:

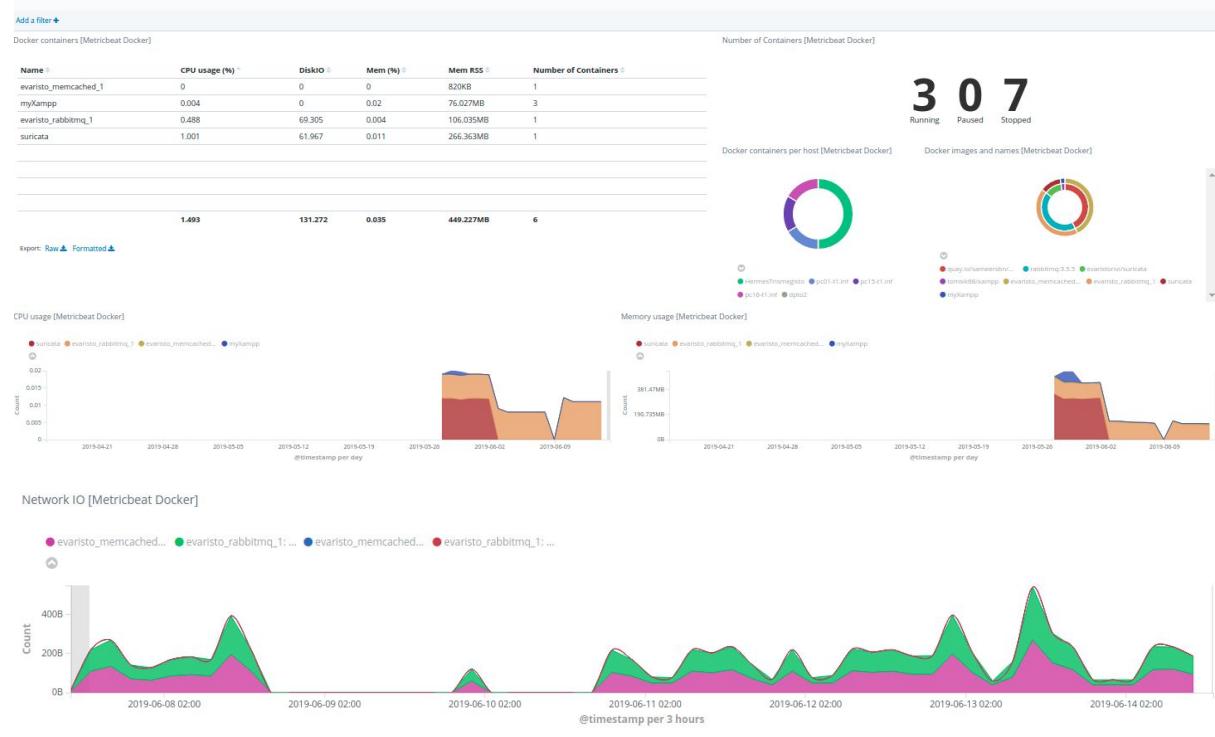




Paneles Metricbeat - módulo apache



Paneles Metricbeat - módulo docker



Configuración de ejemplo de filebeat:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/filebeat/filebeat.yml

Configuración de ejemplo de metricbeat:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/metricbeat/metricbeat.yml

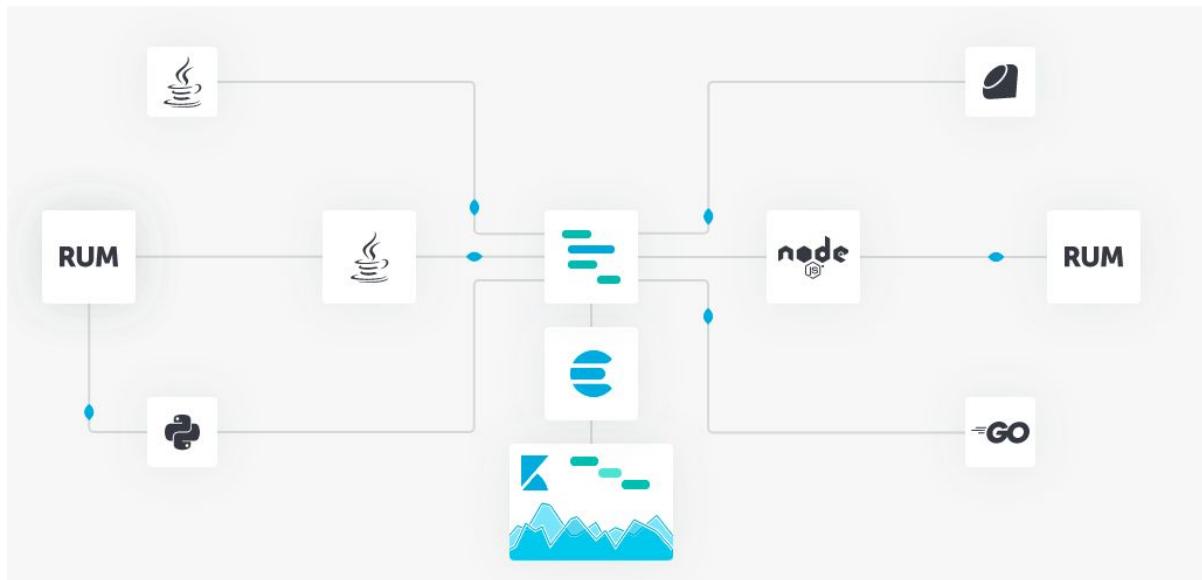
APM

En ELK Stack también pensaron en el APM:

El monitoreo del rendimiento de aplicaciones (APM) es un área de la tecnología de la información (TI) que se centra en asegurarse de que los programas de aplicaciones de software se desempeñan tal y como se espera. El objetivo de la supervisión del rendimiento es proporcionar a los usuarios finales una experiencia de calidad.

Las herramientas de monitoreo de aplicaciones proporcionan a los administradores la información que necesitan para descubrir rápidamente, aislar y resolver problemas que impactan negativamente en el rendimiento de una aplicación. Tales herramientas pueden ser específicas para una aplicación en particular, o monitorizar

varias aplicaciones en la misma red, recogiendo datos sobre el uso del CPU cliente, demandas de memoria, rendimiento de datos y ancho de banda.



El proceso de instalación APM consta de dos partes, por lado instalamos el servidor APM:

<https://www.elastic.co/es/downloads/past-releases#apm-server-oss>

The screenshot shows the 'Past Releases' section for the APM Server OSS 6.7.1. The page includes a search bar with filters for 'Clear All Filters', 'APM Server OSS', and '6.7.1'. Below the search bar, the release information is displayed: 'APM Server OSS 6.7.1' (released April 04, 2019), a 'See Release Notes' link, and a 'Download' button.

```
sudo curl -L -O https://artifacts.elastic.co/downloads/apm-server/apm-server-oss-6.7.1-amd64.deb
sudo dpkg -i apm-server-oss-6.7.1-amd64.deb
```

Configuración de ejemplo

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/apm-server/apm-server.yml

Llegados a este punto podremos recibir información de los agentes APM que deberán de estar integrados en cada aplicación. Kibana nos proporciona varios procedimientos en la sección APM Agents.

APM Agents

Java RUM (JS) Node.js **Django** Flask Ruby on Rails Rack Go

4 Install the APM agent

Install the APM agent for Python as a dependency.

[Copy snippet](#)

```
$ pip install elastic-apm
```

5 Configure the agent

Agents are libraries that run inside of your application process. APM services are created programmatically based on the `SERVICE_NAME`.

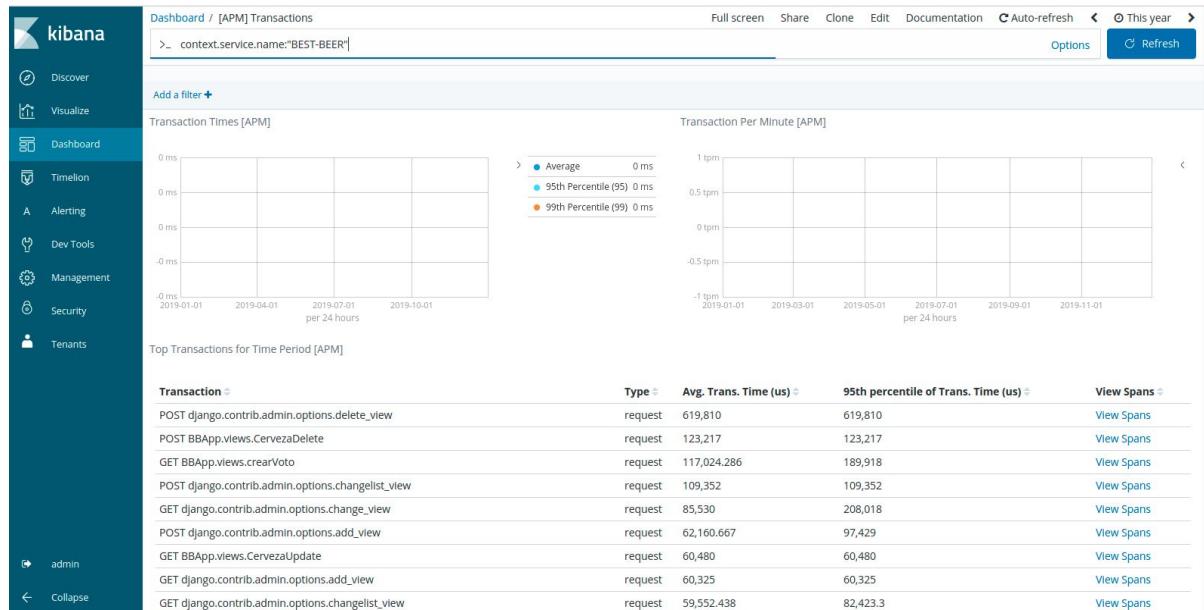
[Copy snippet](#)

```
# Add the agent to the installed apps
INSTALLED_APPS = (
    'elasticapm.contrib.django',
    # ...
)

ELASTIC_APM = {
    # Set required service name. Allowed characters:
    # a-z, A-Z, 0-9, -, _, and space
    'SERVICE_NAME': '',

    # Use if APM Server requires a token
    'SECRET_TOKEN': '',
}
```

Como ejemplo, se ha integrado el agente APM en la aplicación Django Best-Beer implementada durante el curso, y podemos observar los siguientes datos de interés útiles para depurar la aplicación:



The screenshot shows the Kibana interface for the APM Span Details dashboard. The left sidebar has sections for Discover, Visualize, Dashboard, Timelion, Alerting, Dev Tools, Management, Security, and Tenants. Under Admin, there is a 'Collapse' button. The main area shows a table titled 'Spans [APM]' with columns: Time, span.type, span.name, span.duration.us, and span.start.us. The table lists 20 spans from May 28th, 2019, at 09:22:14.358, mostly related to database operations (db) like SELECT and INSERT queries on tables like django_session, auth_user, BBApp_cerveza, and BBApp_votaciones.

Time	span.type	span.name	span.duration.us	span.start.us
May 28th 2019, 09:22:14.358	db	sqlite3.dbapi2.connect	680	-
May 28th 2019, 09:22:14.359	db	SELECT FROM django_session	1,030	-
May 28th 2019, 09:22:14.360	db	SELECT FROM	282	-
May 28th 2019, 09:22:14.362	db	SELECT FROM auth_user	420	-
May 28th 2019, 09:22:14.362	db	SELECT FROM	212	-
May 28th 2019, 09:22:14.363	db	BEGIN	176	-
May 28th 2019, 09:22:14.364	db	SELECT FROM BBApp_cerveza	283	-
May 28th 2019, 09:22:14.364	db	SELECT FROM	193	-
May 28th 2019, 09:22:14.366	db	SELECT FROM BBApp_pub_cerveza	301	-
May 28th 2019, 09:22:14.366	db	SELECT FROM	193	-
May 28th 2019, 09:22:14.368	db	SELECT FROM BBApp_votaciones	313	-
May 28th 2019, 09:22:14.368	db	SELECT FROM	201	-
May 28th 2019, 09:22:14.370	db	INSERT INTO django_admin_log	463	-
May 28th 2019, 09:22:14.370	db	SELECT FROM	249	-
May 28th 2019, 09:22:14.372	db	DELETE FROM BBApp_pub_cerveza	237	-

The screenshot shows the 'Top Errors for Time Period [APM]' visualization. It displays a table with columns: Message, Number of Errors, Type, Culprit, and App Name. The table lists three errors related to 'UnicodeEncodeError' occurring in 'django.utils.encoding.force_text' with 'BEST-BEER' as the app name. Each error has a 'View Error Details' link.

Message	Number of Errors	Type	Culprit	App Name
UnicodeEncodeError: 'ascii' codec can't encode character u'\xe9' in position 5: ordinal not in range(128)	4	UnicodeEncodeError	django.utils.encoding.force_text	BEST-BEER
UnicodeEncodeError: 'ascii' codec can't encode character u'\xe9' in position 5: ordinal not in range(128)	1	UnicodeEncodeError	django.utils.encoding.force_text	BEST-BEER
UnicodeEncodeError: 'ascii' codec can't encode character u'\xe9' in position 5: ordinal not in range(128)	1	UnicodeEncodeError	django.utils.encoding.force_text	BEST-BEER

Concluimos el manual de instalación de Beats. Todos son similar en la instalación, incluídos los Beats de la comunidad³².

Logstash

Descarga

En el siguiente catálogo elegiremos Elasticsearch OSS y la versión acorde a nuestra versión de Elasticsearch.

<https://www.elastic.co/es/downloads/past-releases>

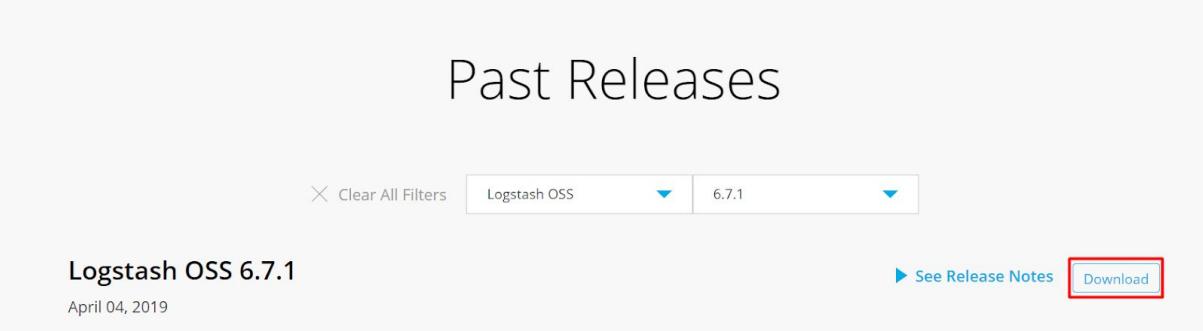
³² Beats de la comunidad: <https://www.elastic.co/guide/en/beats/libbeat/current/community-beats.html>

Past Releases

X Clear All Filters Logstash OSS ▾ 6.7.1 ▾

Logstash OSS 6.7.1 April 04, 2019

▶ See Release Notes **Download**



Hacemos clic derecho encima del enlace que más nos interese, y copiamos el enlace.



Instalación

```
sudo curl -L -O https://artifacts.elastic.co/downloads/logstash/logstash-oss-6.7.1.deb  
sudo dpkg -i logstash-oss-6.7.1.deb
```

Una vez instalado lo arrancamos con:

```
sudo systemctl start logstash
```

Para que se ejecute al inicio:

```
sudo systemctl enable logstash
```

Logstash puede costar al principio arrancarlo, si cometemos errores en los ficheros de configuración, por lo que podremos observar cualquier problema con:

```
sudo tail -f /var/log/logstash/logstash-plain.log
```

El montón de memoria JVM fue configurado para usar 4GB en éste ejemplo en el que la MV contaba con 8 GB:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/logstash/jvm.options

Configuración

En /etc/logstash/conf.d es donde crearemos los ficheros de configuración. Podemos optar por centralizar la configuración en un solo fichero o dividirlo según vemos conveniente. Logstash cargará correctamente la configuración en cualquiera de los dos casos.

Input

Cualquier configuración de Logstash debe contener al menos un plugin de entrada y un plugin de salida. Los filtros son opcionales. Como primer ejemplo de cómo puede ser un fichero de configuración simple, empezaremos con uno que lee una serie de datos de prueba de un fichero y lo envía como salida a la consola en una forma estructurada. Esta es una configuración muy útil al desarrollar una configuración, ya que nos permite iterar rápidamente y construir la configuración. Supondremos que nuestro fichero de configuración se llama test.conf y que está guardado en el directorio “/home/logstash” junto con el fichero que contiene nuestros datos de prueba:

```
input {
  file {
    path => ["/home/logstash/testdata.log"]
    sincedb_path => "/dev/null"
    start_position => "beginning"
  }
}
filter {
}
output {
  stdout {
    codec => rubydebug
  }
}
```

Aquí vemos los tres grupos de nivel principal que forman parte de toda configuración de Logstash: input (entrada), filter (filtro) y output (salida).

En la sección de entrada, especificamos un file input plugin (plugin de entrada de fichero) e ingresamos la ruta de nuestro fichero de datos de prueba a través de la directiva path (ruta). Establecimos la directiva “start_position” en “beginning” para

ordenar al plugin que lea el fichero desde el principio cada vez que se descubra un nuevo fichero.

Si bien el plugin de entrada de fichero de Logstash es una excelente manera para comenzar a desarrollar configuraciones, Filebeat es el producto recomendado para la recopilación y el envío de registros desde servidores host. Filebeat puede enviar registros a Logstash y Logstash puede recibir y procesar estos registros con la entrada de Beats.

Filebeat tiene un rendimiento más optimizado y requiere un menor uso de recursos, lo cual es ideal para ejecutar como agente.

El plugin de salida stdout escribe los datos en la consola y el códec rubydebug ayuda a mostrar la estructura, lo cual simplifica la depuración durante el desarrollo de la configuración.

- Input plugins
 - <https://www.elastic.co/guide/en/logstash/current/input-plugins.html>
- Filter plugins
 - <https://www.elastic.co/guide/en/logstash/current/filter-plugins.html>
- Output plugins
 - <https://www.elastic.co/guide/en/logstash/current/output-plugins.html>
- Codec plugins:
 - <https://www.elastic.co/guide/en/logstash/current/codec-plugins.html>

En un caso real, establecemos como input el plugin beat como vemos a continuación:

00inputbeat.conf

```

1 input {
2   beats {
3     type => "beat"
4     port => 5044
5     ssl => true
6     ssl_certificateAuthorities => ["/etc/metricbeat/MyRootCA.pem"]
7     ssl_certificate => "/etc/metricbeat/odfe-node1.pem"
8     ssl_key => "/etc/metricbeat/odfe-node1.key"
9
10   }
11 }
```

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/logstash/conf.d/00inputbeat.conf

Filter

Pondremos como ejemplo un log de squid, donde podemos ver varios campos que nos interesan:



Al analizar registros de texto, hay especialmente dos filtros que se utilizan con más frecuencia:

- **dissect:**
 - analiza registros de acuerdo con delimitadores.
- **grok:**
 - funciona de acuerdo con la coincidencia de expresiones regulares.

Grok es generalmente más potente y puede manejar una mayor variedad de datos. Sin embargo, la coincidencia de expresiones regulares puede usar más recursos y ser más lenta, en especial si no se optimiza en forma correcta.

Dissect

Al trabajar con el filtro dissect, se especifica una secuencia de campos para extraer además de los delimitadores, los campos. El filtro realiza una sola pasada sobre los datos y busca coincidencias entre los delimitadores en el patrón. Al mismo tiempo, los datos entre los delimitadores se asignan a los campos especificados. El filtro no valida el formato de los datos extraídos.

Los separadores que se usan al analizar estos datos mediante el filtro dissect se resaltan en rosa a continuación.

1524206424.145 106 207.96.0.0 TCP_HIT 200 68247 GET http://elastic.co/guide/en/logstash/current/images/logstash.gif - NONE - image/gif

El primer campo contiene la marca de tiempo y está seguido por uno o más espacios, según la longitud del siguiente campo de duración. Podemos especificar el campo de marca de tiempo como %{timestamp}, pero para que acepte un número variable de espacios como separador, debemos agregar un sufijo -> al campo. Todos los demás separadores en la entrada del registro consisten en sólo un carácter.

Grok

Grok usa patrones de expresiones regulares para hacer coincidir campos y delimitadores. La siguiente figura muestra los campos que deben capturarse en azul y los delimitadores en rojo.

1524206424.145 106 207.96.0.0 TCP_HIT/200 68247 GET http://elastic.co/guide/en/logstash/current/images/logstash.gif - NONE/-image/gif

Grok comenzará a buscar coincidencias entre los patrones configurados desde el principio y continuará hasta mapear todo el evento o hasta determinar que no se puede encontrar una coincidencia. Según el tipo de patrones utilizados, esto puede requerir que grok procese parte de los datos varias veces.

Grok viene con una gran variedad de patrones³³ listos para usar.

Como norma general, conviene consultar éste repositorio, antes de embarcarse en la creación de un patrón personalizado. En este caso, existe un repositorio con el filtro para analizar registros de acceso de Squid³⁴:

SQUID3

```
%{NUMBER:timestamp}\s+%\{NUMBER:duration\}\s%{IP:client_address}\s%{WORD:cache_result}/%\{POSINT:status_code\}\s%{NUMBER:bytes}\s%{WORD:request_method}\s%{NOTSPACE:url}\s(%{NOTSPACE:user}|-)\s%{WORD:hierarchy_code}/%\{IP:ORHOST:server\}\s%{NOTSPACE:content_type}
```

Al crear una configuración grok, hay una serie de patrones estándar que se usan comúnmente:

- WORD (palabra):
 - patrón que hace coincidir una sola palabra.
 - NUMBER (número):
 - patrón que hace coincidir un entero positivo o negativo o un número de punto flotante.
 - POSINT:
 - patrón que hace coincidir un entero positivo..
 - IP:
 - patrón que hace coincidir una dirección IP IPv4 o IPv6.
 - NOTSPACE:

33 Patrones Grok:

<https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/grok-patterns>

³⁴ Repositorio con patrones Grok para squid:

<https://github.com/logstash-plugins/logstash-patterns-core/blob/master/patterns/squid>

- patrón que hace coincidir cualquier cosa que no sea un espacio.
- SPACE:
 - patrón que hace coincidir cualquier cantidad de espacios consecutivos.
- DATA:
 - patrón que hace coincidir una cantidad limitada de cualquier tipo de datos.
- GREEDYDATA:
 - patrón que hace coincidir todos los datos restantes.

Finalmente así quedó nuestra configuración para squid:

10squid.conf

```

1 filter {
2   if [source] == "/var/log/squid/access.log" {
3     grok {
4       match => ["message", "%{INT:timestamp} %{INT}\s*%{NUMBER:request_msec:float} %{IPORHOST:src_ip} %{WORD:cache_result}/%{
5       add_tag => ["squid"]
6     }
7
8     date {
9       match => [ "timestamp", "UNIX" ]
10    }
11
12
13  }
14 }
15

```

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/ogstash/conf.d/10squid.conf

Estos son los patrones que usaremos al construir nuestra configuración del filtro grok. La manera de crear configuraciones de grok es, por lo general, empezar desde la izquierda y construir gradualmente el patrón, capturando el resto de los datos con un patrón GREEDYDATA.

Pondremos como ejemplo el filtro Grok que tuvimos que customizar para recoger cierta información de los usuarios que iniciaban sesión de manera gráfica:

Nos fue muy útil apoyarnos en herramientas de depuración Grok como:
<http://grokdebug.herokuapp.com/>

The screenshot shows the Grok Debugger interface. At the top, there are tabs: 'Grok Debugger' (selected), 'Debugger', 'Discover', and 'Patterns'. Below the tabs, a log entry is displayed in red: "May 30 09:07:12 pc16-t1 systemd: pam_unix(systemd-user:session): session closed for user antonioroqa by (uid=0)". Below the log entry, a Grok pattern is shown: "%{SYSLOGTIMESTAMP} %{SYSLOGHOST} systemd: pam_unix\\(systemd-user:session\\): session %{WORD:estado} for user %{USERNAME:username} by \\(uid=%{INT:uid:int}\\)". At the bottom of the interface, there are several checkboxes: 'Add custom patterns', 'Keep Empty Captures', 'Named Captures Only', 'Singles', 'Autocomplete', and a 'Go' button.

En la zona superior introduciremos la línea de log que queremos trabajar y debajo vamos construyendo el patrón Grok.

Debajo de éste veremos la respuesta JSON, en caso de no encontrar errores.

The screenshot shows the Grok Debugger interface with the resulting JSON structure. The JSON object contains several arrays corresponding to different log fields: 'SYSLOGTIMESTAMP', 'MONTH', 'MONTHDAY', 'TIME', and 'HOUR'. Each array contains a single element: "May 30 09:07:12", "May", "30", "09:07:12", and "" respectively.

Nuestro filtro grok customizado quedó de la siguiente manera:

```
%{SYSLOGTIMESTAMP} %{SYSLOGHOST} systemd: pam_unix\\(systemd-user:session\\): session
%{WORD:estado} for user %{USERNAME:username} by \\(uid=%{INT:uid:int}\\)"
```

y fue integrado dentro del filtro System.conf que veremos a continuación.

System.conf

Las siguientes tuberías³⁵ con filtros grok controlan todos los eventos de auth y syslog:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/logstash/conf.d/system.conf

ldap.conf

Para tratar los campos referentes al LDAP del departamento se utilizaron los siguientes filtros Grok:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/logstash/conf.d/ldap.conf

apache.conf

Para tener un mayor control del apache que corre en el departamento se utilizaron los siguientes filtros Grok:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/logstash/conf.d/apache.conf

89addpersonalizICO.conf

Se nos planteó la necesidad de representar la organización interna del departamento, la cual es:

- 192.168.3.0 => es ICO 1
- 192.168.2.0 => es ICO 2
- 192.168.1.0 => es ICO 3
- 192.168.0.0 => es ICO 4

Del x.x.x.1 al x.x.x.116 están fichados, a partir del 120 no, serán portátiles o máquinas virtuales

Lo logramos con el siguiente filtro Grok, totalmente customizado:

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/logstash/conf.d/89addpersonalizICO.conf

Fue importante contar con herramientas³⁶ para ayudarnos con las expresiones regulares.

³⁵ Tuberías para los módulos de Filebeat:

<https://www.elastic.co/guide/en/logstash/6.7/logstash-config-for-filebeat-modules.html>

³⁶ Herramientas para expresiones regulares:

<https://www.regexpal.com/>

<https://www.analyticsmarket.com/freetools/ipregex/>

Output

Según el tipo de log le añadiremos un nombre de índice y lo enviaremos al servidor Elasticsearch, adjuntando la información SSL y de autenticación.

99out.conf

Aquí definimos la salida, dependiendo del source, o de la tag que nos interese, lo enviaríamos a Elasticsearch con un nombre de índice u otro.

https://github.com/evaristorivi/Proyecto_final-ELK-Stack-Opendistro/blob/master/etc/logstash/conf.d/99out.conf

4.3 Manual de administración.

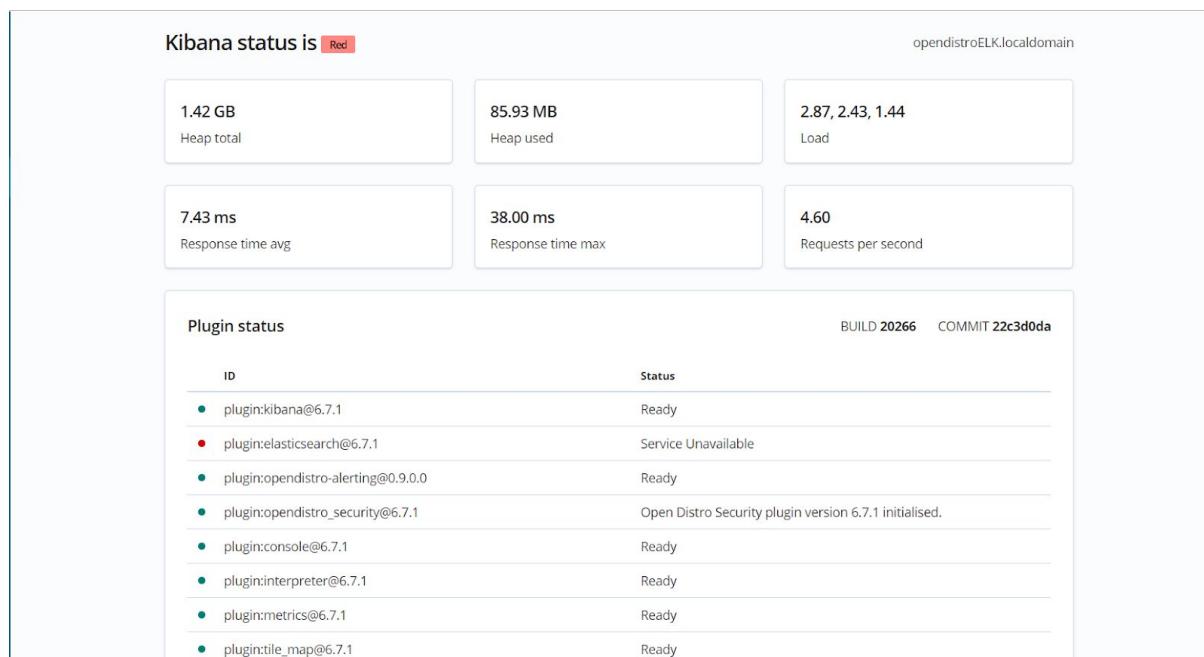
A continuación se expondrán procedimientos, distintos al proceso de “Instalación” o “Uso” de la aplicación. Tareas de administración como correcciones de errores, actualizaciones, etc..

Levantar Elasticsearch tras un apagado

Es común, que al levantar Elasticsearch tarde un largo rato en inicializarse completamente, y hasta que esto suceda, veremos varios mensajes de ERROR en el log de Elasticsearch, como se muestra a continuación:

c.a.o.s.a.BackendRegistry

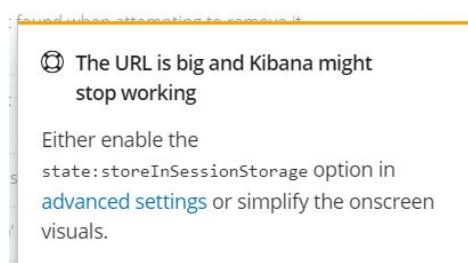
```
[2019-05-22T01:48:50,289][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:50,293][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:51,128][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:53,645][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:55,721][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:55,725][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:55,726][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:56,162][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
[2019-05-22T01:48:56,333][ERROR][c.a.o.s.a.BackendRegistry] [uUzpDuY] Not yet initialized (you may need to run securityadmin)
```



Deberemos de esperar, alrededor de veinte minutos o incluso media hora. a que se inicialice por completo.

Errores en Kibana con url demasiado largas

Llegado el momento, puede darse la situación de que hayamos concatenado una gran cantidad de filtros, y por tanto se nos haya formado una url demasiado larga, provocando un error.



Tiene solución, hay que hacer clic en el hipervínculo del mensaje que aparecerá en ese momento, y en la página de configuración activar lo que se indica.

Almacenar URLs en almacenamiento de sesión

La URL a veces puede llegar a ser demasiado grande para que algunos navegadores la manejen. Para contrarrestar esto, estamos probando si el almacenamiento de partes de la URL en el almacenamiento de sesión podría ayudar. ¡Por favor, hágnos saber cómo le va!

Defecto: `false`

estado: `storeInSessionStorage`
 Apagado
[Restablecen a los predeterminados](#)

Procedimiento de actualización de versión

En el momento de comenzar la investigación instalamos la versión de Elasticsearch 6.7.1 con la versión 0.8.0 de Open Distro for Elasticsearch

Version history

Open Distro for Elasticsearch version	Release highlights	Elasticsearch version
0.9.0	Bumps Elasticsearch version.	6.7.1
0.8.0	Bumps Elasticsearch version.	6.6.2
0.7.1	Fixes Kibana multitenancy.	6.5.4
0.7.0	Initial release.	6.5.4

Semanas después salió la nueva actualización la 6.7.1 con 0.9.0 de OpenDistro for Elasticsearch y procedimos al update, dejando registrado todos los errores en <https://discuss.opendistrocommunity.dev/t/update-to-version/662>

Con sudo yum update aparecían los siguientes errores:

→ Resolución de dependencias finalizada

Error: Paquete: opendistro-performance-analyzer-0.9.0.0-1.noarch (opendistroforelasticsearch-artifacts-repo)

Necesita: elasticsearch-oss = 6.7.1

La solución era ejecutar los siguientes comandos:

sudo yum update opendistroforelasticsearch

sudo yum update opendistroforelasticsearch-kibana

Una vez hecho esto:

sudo systemctl daemon-reload

Pero al ver el log de Elasticsearch observamos que no arrancaba, y aparecían los siguientes errores.

Errores

java.lang.IllegalStateException: Duplicate key ingest-user-agent

java.lang.IllegalStateException: Duplicate key purge ingest-geoip

La razón está documentada aquí:

https://www.elastic.co/guide/en/elasticsearch/reference/6.7/breaking-changes-6.7.html#breaking_67_plugin_changes

Por lo tanto la solución fue ejecutar:

/usr/share/elasticsearch/bin/elasticsearch-plugin remove --purge ingest-geoip

/usr/share/elasticsearch/bin/elasticsearch-plugin remove --purge ingest-user-agent

Una vez hecho esto, todo estaba funcionando perfectamente.

The screenshot shows the Kibana 6.7.1 management interface. The left sidebar has a dark blue background with white icons and text. The 'Management' link is highlighted with a blue background. The main panel has a light gray background with a central box containing a gear icon and the text 'Kibana 6.7.1 management' and 'Manage your indices, index patterns, saved objects, Kibana settings, and more.' Below this is a note: 'A full list of tools can be found in the left menu.'

```
[evaristo@opendistroELK ~]$ curl -XGET https://localhost:9200 -u admin:admin --insecure
{
  "name" : "uUzpDuY",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Rp0TvsyQ_Ss_MJcdv1J9g",
  "version" : {
    "number" : "6.7.1",
    "build_flavor" : "oss",
    "build_type" : "rpm",
    "build_hash" : "2f32220",
    "build_date" : "2019-04-02T15:59:27.961366Z",
    "build_snapshot" : false,
    "lucene_version" : "7.7.0",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
[evaristo@opendistroELK ~]$ curl -XGET https://localhost:9200/_cat/nodes?v -u admin:admin --insecure
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
192.168.1.20      82        97     38   0.31   0.63   1.00 mdi      * uUzpDuY
[evaristo@opendistroELK ~]$ curl -XGET https://localhost:9200/_cat/plugins?v -u admin:admin --insecure
name      component      version
uUzpDuY opendistro_alerting 0.9.0.0
uUzpDuY opendistro_performance_analyzer 0.9.0.0
uUzpDuY opendistro_security 0.9.0.0
uUzpDuY opendistro_sql 0.9.0.0
[evaristo@opendistroELK ~]$
```

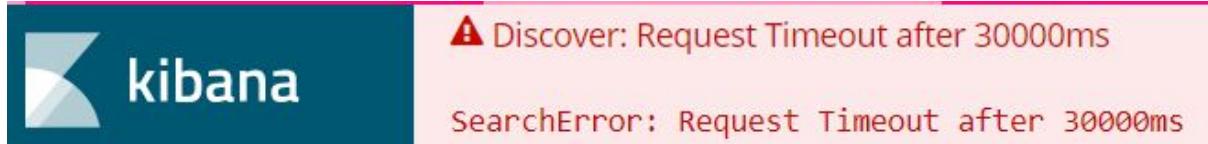
Se ha propuesto³⁷ en la comunidad de Opendistro, traer para futuras actualizaciones un “auto upgrade” similar al que ya tiene la última versión de Elastic.

The screenshot shows a GitHub discussion thread under the 'Open Distro for Elasticsearch' repository. The title of the thread is 'Upgrade Assistant'. The first comment is from user 'evaristorivi' asking if there are plans to implement an upgrade assistant. User 'carlmead' responds positively, mentioning they have created an issue on GitHub. The GitHub issue link is provided: <https://github.com/opendistro-for-elasticsearch/community/issues/699>. The issue details a Kibana UI for validation checks on an existing 6.7 domain against 7.x compatibility. The conversation continues with 'evaristorivi' thanking 'carlmead'.

³⁷ Propuesta del autor de éste proyecto, en la comunidad de Opendistro:
<https://discuss.opendistrocommunity.dev/t/upgrade-assistant/699>

Request Timeout after 300000ms

Es probable que nos encontremos con éste error en Kibana:



Para solucionarlo ampliamos el tiempo de espera en /etc/kibana/kibana.yml añadiendo:

```
elasticsearch.requestTimeout: 90000
```

y reiniciamos el proceso de Kibana.

```
systemctl restart kibana
```

De seguir apareciendo, la solución sería aumentar recursos de memoria. En este caso la MV Java estaba limitada a 1GB de RAM, por lo que es comprensible que al ejecutar la consulta para cierto volumen de datos, ocurra esto. Se amplió a 4GB y el mensaje dejó de aparecer.

Estado del clúster

Contamos con la herramienta PerfTop CLI, que descargamos desde:

<https://opendistro.github.io/for-elasticsearch/downloads.html>

Veremos cómo ejecutar cada uno de sus paneles:

Error Filebeat exiting

En una de las múltiples pruebas con logstash, tras reiniciarlo, dejamos de recibir datos de squid. Analizando el log de Filebeat del equipo donde estaba corriendo squid, pudimos observar que Filebeat estaba indicando:

exiting: can onlye start an input when all relared stated are finished

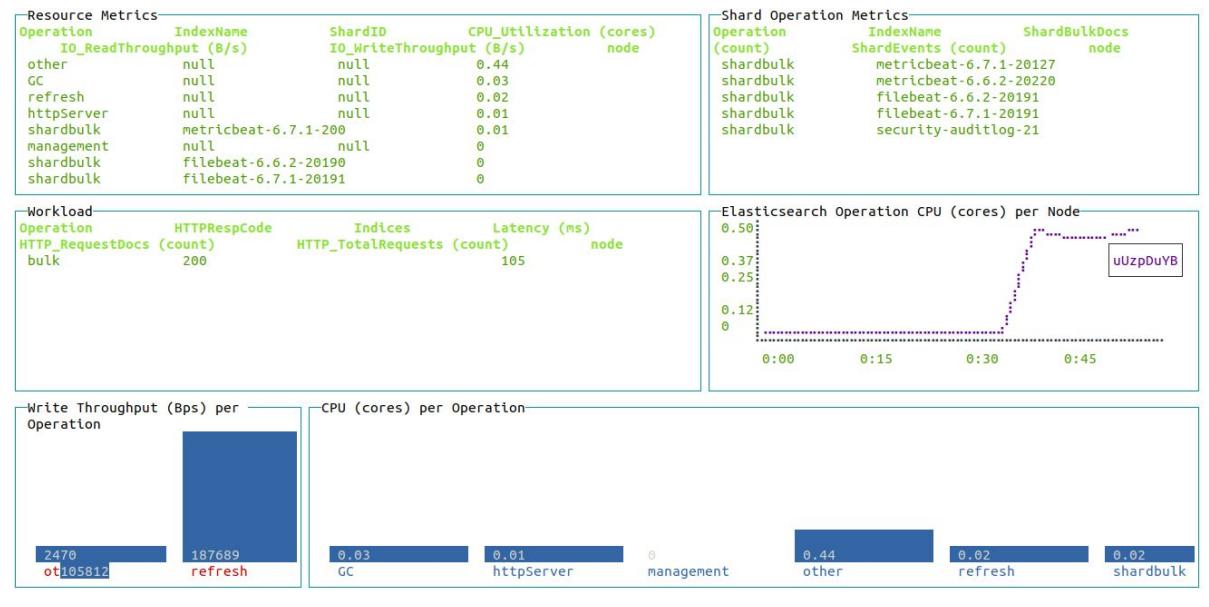
Por lo que hemos podido comprobar, coincidió la rotación del fichero de squid con el reinicio de logstash. La solución fue eliminar el fichero de log:

/var/log/squid/access.log

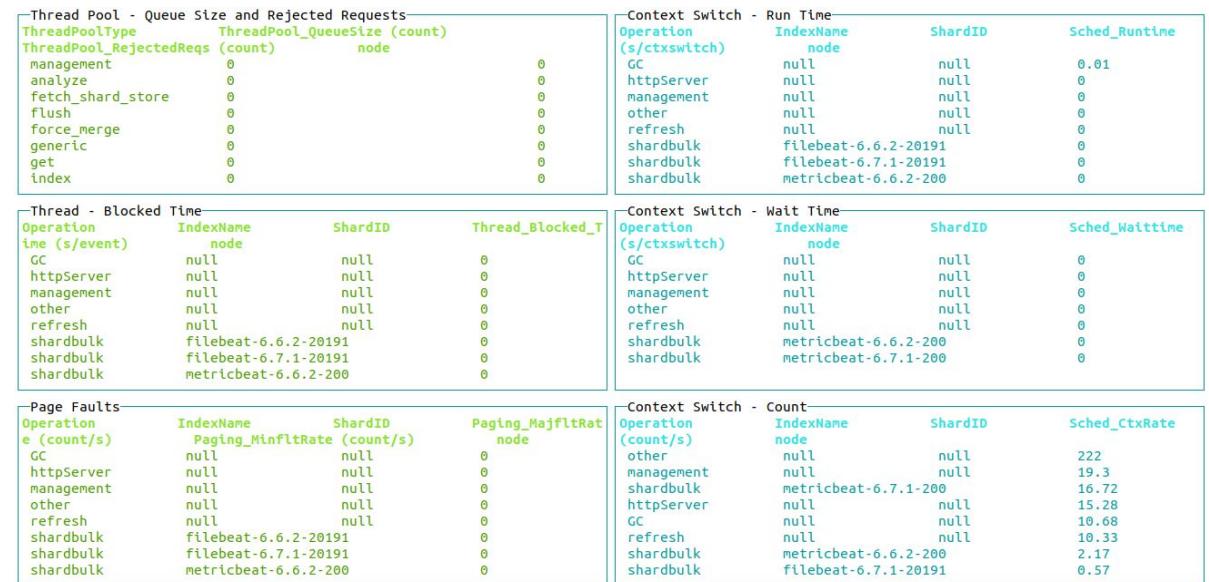
y reiniciar los servicios squid y logstash, en ese mismo orden.

PerfTop CLI

./perf-top-linux --dashboard ClusterOverview --endpoint 192.168.1.20:9600



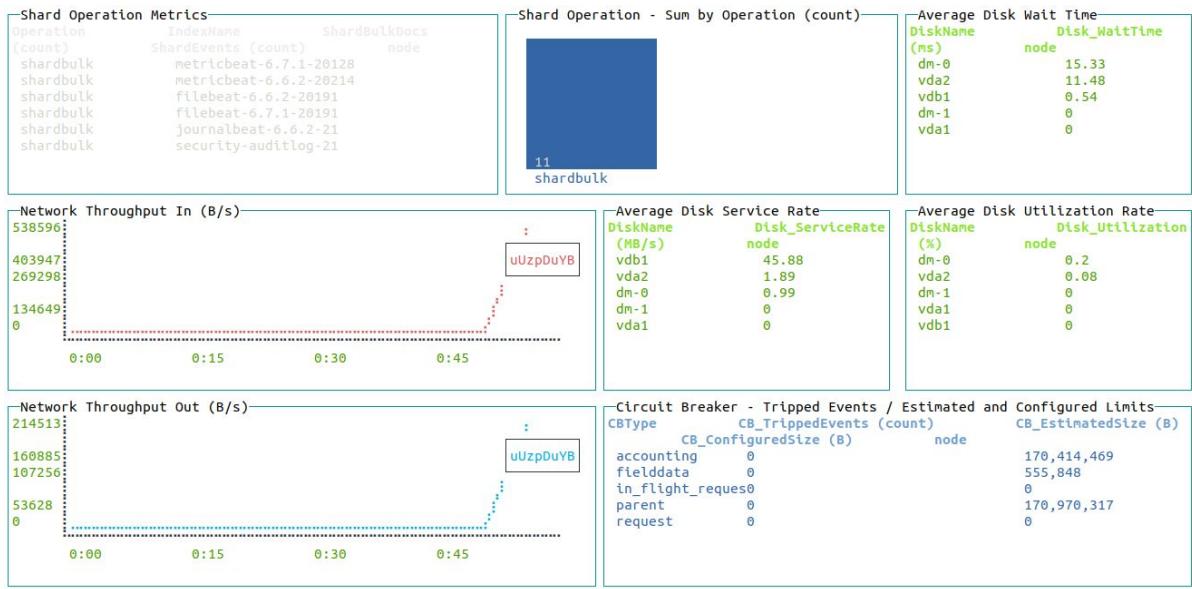
./perf-top-linux --dashboard ClusterThreadAnalysis --endpoint 192.168.1.20:9600



./perf-top-linux --dashboard NodeAnalysis --endpoint 192.168.1.20:9600



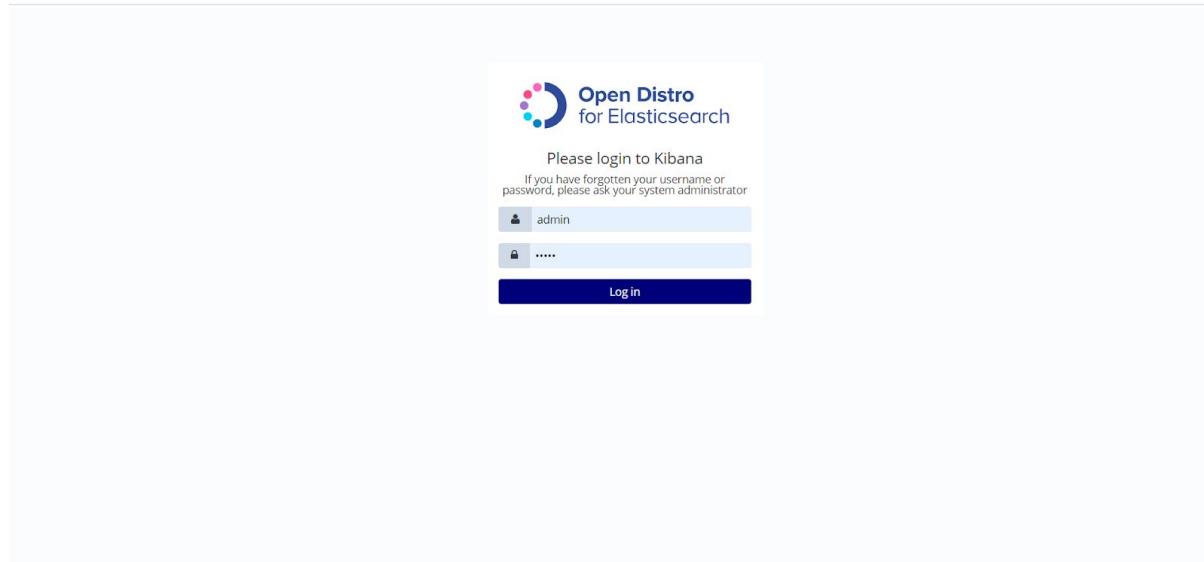
./perf-top-linux --dashboard ClusterNetworkMemoryAnalysis --endpoint 192.168.1.20:9600



De ésta forma podremos monitorizar nuestro clúster de Elasticsearch.

4.4 Manual de usuario.

Accedemos a la dirección del servidor Kibana
http://SERVIDOR_KIBANA:9595



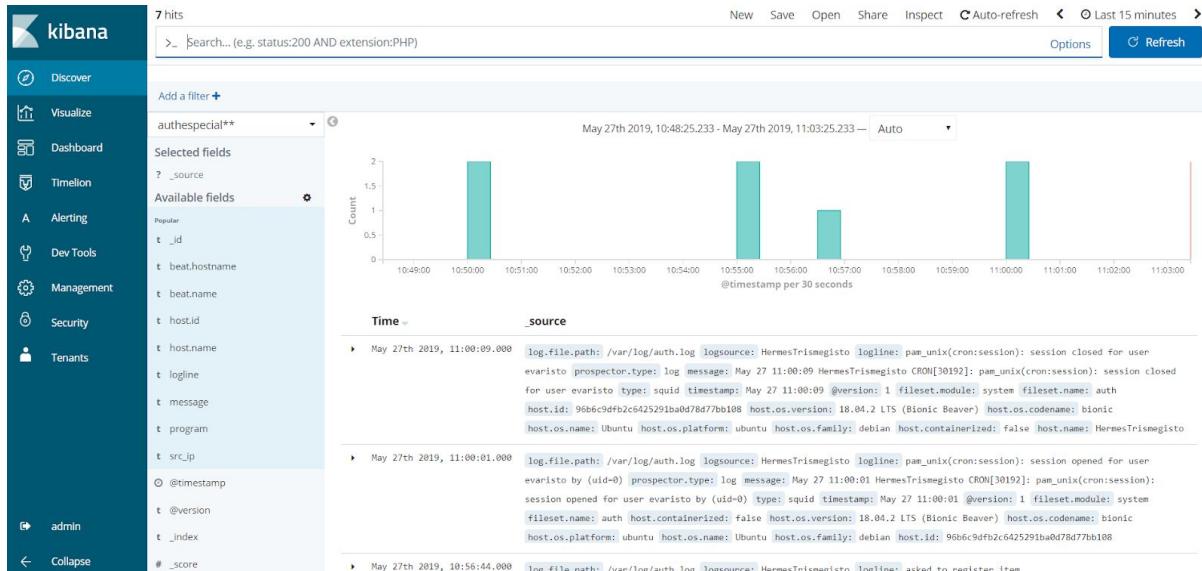
El usuario por defecto es:

Usuario: admin

Contraseña: admin

Una vez dentro veremos la pantalla principal, en la que tenemos los accesos directos que ya comentamos en el manual de instalación de Filebeat.

Discover



En la pestaña Discover podremos hacer todo tipo de consultas sobre los datos que nos lleguen.

Discover nos permitirá explorar los datos que nos lleguen al servidor Elasticsearch.

Tiene acceso a cada documento en cada índice que coincide con el patrón de índice seleccionado. Podemos enviar consultas de búsqueda, filtrar los resultados de la búsqueda y ver datos de documentos. También podemos ver el número de documentos que coinciden con la consulta de búsqueda y obtener estadísticas de valor de campo.

Si se configura un campo de tiempo (@timestamp) para el patrón de índice seleccionado, la distribución de documentos a lo largo del tiempo se muestra en un histograma en la parte superior de la página.

Podemos configurar el intervalo de actualización en la siguiente sección.



Aquí podemos configurar el segmento temporal que nos interesa obtener:

En la pestaña “Visualize” tenemos todas las gráficas ya creadas.

Podemos establecer un filtro de tiempo si el índice contiene eventos basados en el tiempo y se configura un campo de tiempo para el patrón de índice seleccionado.

Por defecto, el filtro de tiempo se establece en los últimos 15 minutos. Podemos usar el selector de tiempo para cambiar el filtro de tiempo, o seleccionar un intervalo de tiempo o rango de tiempo específico en el histograma en la parte superior de la página.

Filtrado con el selector de tiempo de edición.

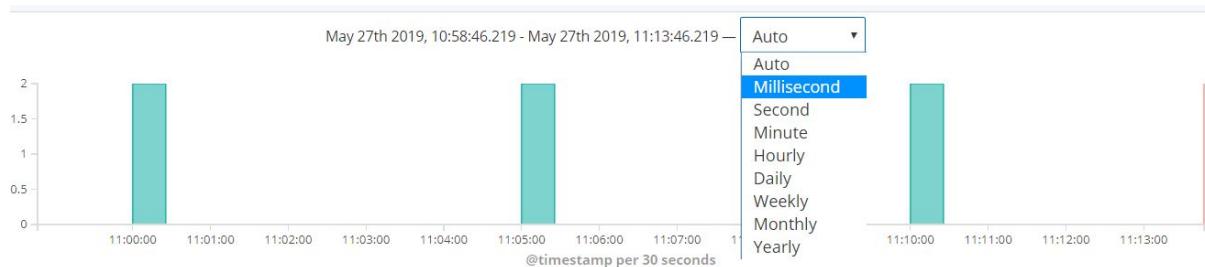
Podemos especificar un filtro de tiempo de una de las siguientes cuatro maneras:

- Quick:
 - Elegimos el periodo deseado.
- Relative:
 - Especificamos un filtro de tiempo relativo a la hora actual.
- Absolute:
 - Especificamos las horas de inicio y finalización del filtro de tiempo.
- Recent:
 - Hacemos clic en uno de los tiempos de la lista de filtros de tiempo utilizados recientemente.

Filtrado desde la edición del histograma.

Podemos configurar un filtro de tiempo desde el histograma de una de estas dos maneras:

- Hacemos clic en la barra que representa el intervalo de tiempo que deseamos ampliar.
- Hacemos clic y arrastramos para ver un período de tiempo específico. Debemos comenzar la selección con el cursor sobre el fondo del gráfico: el cursor cambia a un signo “+” cuando nos desplazamos sobre un punto de inicio válido.



Utilizaremos el botón Atrás del navegador para deshacer los cambios.

Búsquedas

Podemos buscar los índices que coinciden con el patrón de índice actual ingresando los criterios de búsqueda en la barra de consultas. De forma predeterminada, podemos usar el lenguaje de consulta estándar de Kibana, que cuenta con autocompletado y una sintaxis simple y fácil de usar. El lenguaje de consulta heredado de Kibana (basado en la sintaxis de consulta de Lucene) todavía está disponible por el momento bajo el menú de opciones en la barra de consultas. Cuando se selecciona este lenguaje de consulta heredado, también se puede usar el DSL de consulta Elasticsearch completo basado en JSON.

Cuando enviamos una solicitud de búsqueda, el histograma, la tabla Documentos y la lista de Campos se actualizan para reflejar los resultados de la búsqueda. El número total de visitas (documentos coincidentes) se muestra en la barra de herramientas. La tabla Documentos muestra los primeros quinientos hits. De forma predeterminada, los resultados se enumeran en orden cronológico inverso, con los documentos más recientes mostrados primero. Podemos invertir el orden de clasificación haciendo clic en el encabezado de la columna Tiempo. También podemos ordenar la tabla por los valores en cualquier campo indexado.

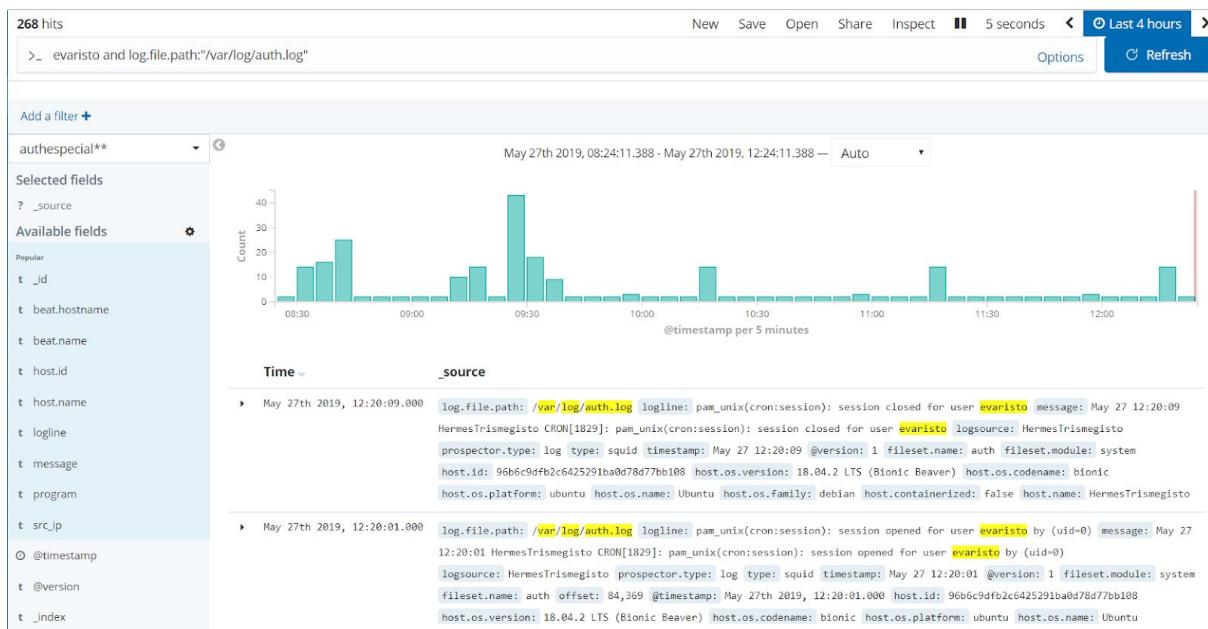
Time	_source
May 27th 2019, 11:15:11	log.file.path: /var/log/auth.log logsource: HermesTrismegisto logline: pam_unix(cron:session): session closed for user evaristo prospector.type: log message: May 27 11:15:11 HermesTrismegisto CRON[30991]: pam_unix(cron:session): session closed for user evaristo type: squid timestamp: May 27 11:15:11 @version: 1 filesset.module: system filesset.name: auth offset: 81,072 @timestamp: May 27th 2019, 11:15:11.000 host.id: 96b6c9dfb2c6425291ba0d78d77bb108 host.containerized: false host.os.codename: bionic host.os.version: 18.04.2 LTS (Bionic Beaver) host.os.name: Ubuntu host.os.platform: ubuntu
May 27th 2019, 11:15:01.000	log.file.path: /var/log/auth.log logsource: HermesTrismegisto logline: pam_unix(cron:session): session opened for user evaristo by (uid=0) prospector.type: log message: May 27 11:15:01 HermesTrismegisto CRON[30991]: pam_unix(cron:session): session opened for user evaristo by (uid=0) type: squid timestamp: May 27 11:15:01 @version: 1 filesset.module: system filesset.name: auth host.id: 96b6c9dfb2c6425291ba0d78d77bb108 host.os.codename: bionic host.os.version: 18.04.2 LTS (Bionic Beaver) host.os.platform: ubuntu host.os.name: Ubuntu host.os.family: debian host.containerized: false
May 27th 2019, 11:10:05.000	log.file.path: /var/log/auth.log logsource: HermesTrismegisto logline: pam_unix(cron:session): session closed for user

Kibana Query Language

Ejemplos:

- response:200 coincidirá con los documentos donde el campo de respuesta coincide con el valor 200.
- Las comillas alrededor de un término de búsqueda iniciarán una búsqueda de frase. Por ejemplo, message:"zorro marrón rápido" buscará la frase "zorro marrón rápido" en el campo de mensaje. Sin las comillas, la consulta se dividirá en tokens a través del analizador configurado del campo de mensajes y coincidirá con los documentos que contienen esos tokens, independientemente del orden en que aparezcan. Esto significa que los documentos con "zorro marrón rápido" coincidirán, pero también lo hará "zorro rápido marrón". Hay que utilizar comillas si desea buscar una frase.
- Los términos de búsqueda múltiples deben estar separados por operadores booleanos explícitos.
 - Los operadores booleanos no distinguen entre mayúsculas y minúsculas.
- AND
 - response:200 and extension:php. Esto coincidirá con los documentos donde la respuesta coincide con 200 y la extensión coincide con php.
 -
- OR
 - response:200 or extension:php coincidirá con los documentos donde la respuesta coincide con 200, la extensión coincide con php, o ambos.
- Por defecto, and tiene una precedencia más alta que or.
 - response: 200 and extension:php or extension:css coincidirá con los documentos donde la respuesta es 200 y la extensión es php O los documentos donde la extensión es css y la respuesta es cualquier cosa.
- Podemos anular la prioridad predeterminada con la agrupación.
 - response:200 and (extension:php or extension:css) coincidirá con los documentos donde la respuesta es 200 y la extensión es php o css.
- Existe una abreviatura que nos permite buscar fácilmente un solo campo para varios valores.
 - response:(200 or 404) busca documentos donde el campo response coincide con 200 o 404. También podemos buscar documentos con campos de múltiples valores que contengan una lista de términos, por ejemplo:tags:(success and info and security)
- Los términos pueden invertirse prefijándolos con not.
 - not response:200 coincidirá con todos los documentos donde la respuesta no es 200.
- También se pueden invertir grupos enteros.

- response:200 and not (extension:php or extension:css)
- Los rangos son similares a lucene con una pequeña diferencia sintáctica.
 - En lugar de bytes:>1000, omitimos los dos puntos: bytes > 1000.
 - >, >=, <, <= Todos son operadores de rango válidos.
- Las consultas de comodines están disponibles. machine.os:win*coincidiría con los documentos donde el campo machine.os comienza con "win", que coincidirá con valores como "windows 7" y "windows 10".



Guardando y abriendo búsquedas

Guardando y abriendo búsquedas

Guardar búsquedas nos permite volver a cargarlas en Discover y usarlas como base para las visualizaciones . Guardar una búsqueda guarda tanto la cadena de consulta de búsqueda como el patrón de índice seleccionado actualmente.

Para guardar la búsqueda actual haremos clic en Guardar en la barra de herramientas de Kibana.

New Save Open Share Inspect

Ingresamos un nombre para la búsqueda y hacemos clic en Guardar .

The screenshot shows the Kibana Discover interface. At the top, it says "18 hits" and has a search bar with placeholder text "Search... (e.g. status:200 AND extension:PHP)". Below the search bar are filter and visualization options. The main area displays a histogram of event counts over time, with a tooltip indicating "May 27th 2019, 11:14:17.260 - May 27th 2019, 11:29:17.260 — Auto". A modal window titled "Save search" is overlaid, asking for a "Title" (with "New Saved Search" entered) and providing "Cancel" and "Confirm Save" buttons.

Podemos importar, exportar y eliminar búsquedas guardadas desde Gestión/Kibana/Objetos guardados.

The screenshot shows the Kibana Management interface under the "Saved Objects" section. It lists several saved objects with their titles and types. Each entry has a checkbox, a "Type" dropdown, a "Delete" button, and an "Export" button. The objects listed include "[Metricbeat HAProxy] HTTP frontend", "[Metricbeat HAProxy] Backend", "[Metricbeat Kubernetes] Overview", "[Metricbeat System] Host overview", "[Metricbeat Kafka] Overview", and "[Packetbeat] PostgreSQL performance".

Abrir una búsqueda guardada editar

Para cargar una búsqueda guardada en Discover:

Haremos clic en “Abrir” en la barra de herramientas de Kibana.
y seleccionaremos la búsqueda deseada.

The screenshot shows a search interface titled "Open Search". At the top right is a close button (X). Below it is a search bar with a magnifying glass icon and the placeholder "Search...". To the right of the search bar is a blue button labeled "Manage searches". The main area displays a list of search queries under the heading "Title". The queries listed are:

- Socket Connects [Auditbeat Auditd]
- Socket Accept / Recvfrom [Auditbeat Auditd]
- Audit Event Table [Auditbeat Auditd]
- DHCPv4 [Packetbeat]
- systemd usuarios logins
- systemd y sshd NUEVO ACCESOS v2
- File Integrity Events [Auditbeat File Integrity]
- Socket Binds [Auditbeat Auditd]
- Process Executions [Auditbeat Auditd]
- nfs

Below the list, there is a "Rows per page" dropdown set to 10, followed by a navigation bar with page numbers 1 through 9.

Si la búsqueda guardada está asociada con un patrón de índice diferente al seleccionado actualmente, al abrir la búsqueda guardada se cambia el patrón de índice seleccionado.

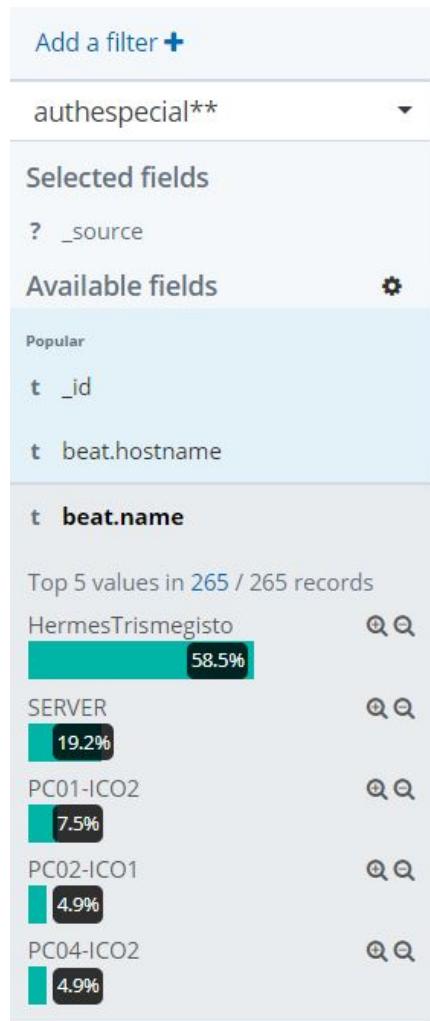
Filtrado por campo

Podemos filtrar los resultados de la búsqueda para mostrar solo aquellos documentos que contienen un valor particular en un campo. También podemos crear filtros negativos que excluyan documentos que contengan el valor de campo especificado.

Podemos agregar filtros de campo de la lista Campos, la tabla Documentos o agregando manualmente un filtro. Además de crear filtros positivos y negativos, la tabla Documentos nos permite filtrar si un campo está presente o no. Los filtros aplicados se muestran debajo de la barra de consultas. Los filtros negativos se muestran en rojo.

Para agregar un filtro de la lista de Campos:

Hacemos clic en el nombre del campo que deseamos filtrar. Esto muestra los cinco valores principales para ese campo.

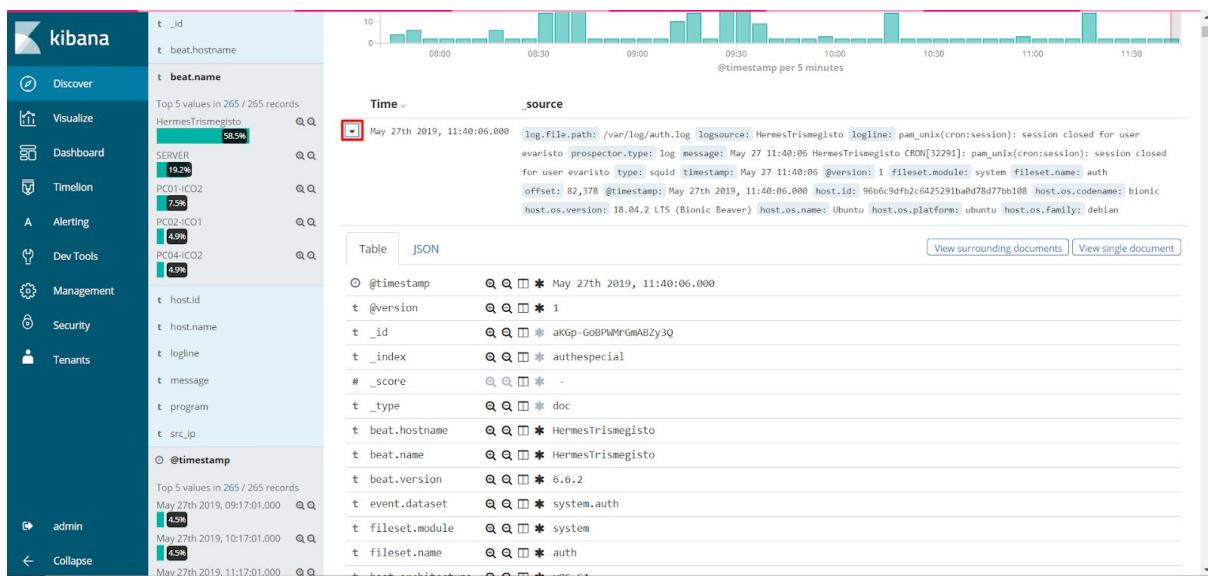


Para agregar un filtro positivo, haremos clic en el botón Filtro positivo . Esto incluye sólo aquellos documentos que contienen ese valor en el campo.

Para agregar un filtro negativo, haremos clic en el botón Filtro negativo . Esto excluye los documentos que contienen ese valor en el campo.

Para agregar un filtro de la tabla Documentos:

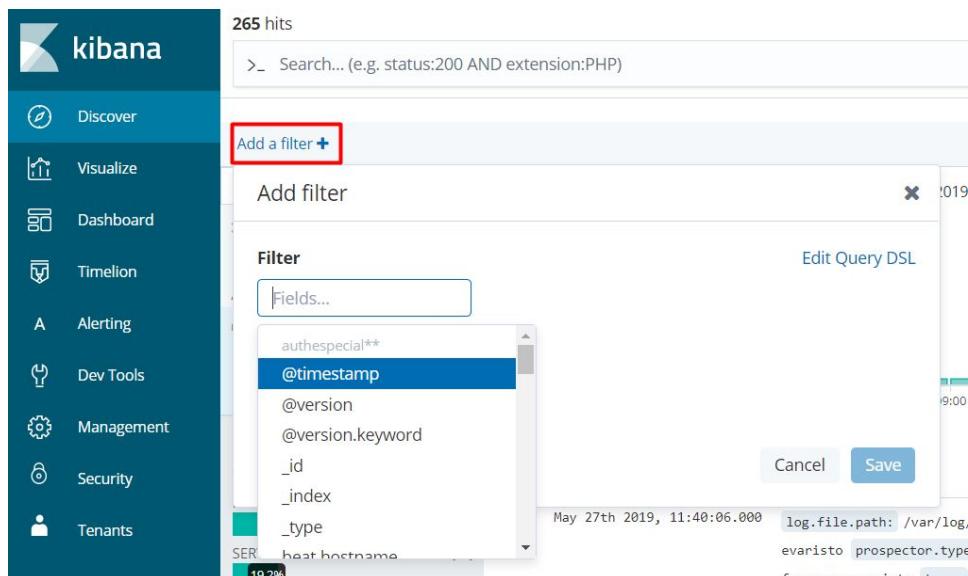
Expandimos un documento en la tabla Documentos haciendo clic en el botón Expandir a la izquierda de la entrada de la tabla del documento.



1. Para agregar un filtro positivo, haremos clic en el botón Filtro positivo a la derecha del nombre del campo. Esto incluye sólo aquellos documentos que contienen ese valor en el campo.
2. Para agregar un filtro negativo, haremos clic en el botón Filtro negativo a la derecha del nombre del campo. Esto excluye los documentos que contienen ese valor en el campo.
3. Para filtrar si los documentos contienen o no el campo, haremos clic en el botón Existe a la derecha del nombre del campo. Esto incluye sólo aquellos documentos que contienen el campo.

Para agregar manualmente un filtro:

Haremos clic en Agregar filtro. Se mostrará una ventana emergente para crear el filtro.



Elegimos un campo para filtrar. Esta lista de campos incluirá los campos del patrón de índice con el que está consultando actualmente.

Luego elegimos una operación para el filtro.

The screenshot shows a 'Filter' configuration window. On the left, there's a dropdown for 'host.architecture' and a button for 'Operators...'. Below that is a 'Label' field with 'Optional' and a 'Typeahead' input. A modal window is open over the main interface, listing filter operators: 'is' (selected), 'is not', 'is one of', 'is not one of', 'exists', and 'does not exist'. At the bottom right of the modal are 'Cancel' and 'Save' buttons. The background shows a log entry with timestamp '11:40:06.000' and file path 'log.file.path: /var/le'.

Se pueden seleccionar los siguientes operadores:

- is
 - Filtrar donde el valor para el campo coincide con el valor dado.
- is not
 - Filtrar donde el valor del campo no coincide con el valor dado.
- is one of
 - Filtre donde el valor del campo coincide con uno de los valores especificados.
- is not one of
 - Filtre donde el valor del campo no coincide con ninguno de los valores especificados.
- is between
 - Filtrar donde el valor para el campo está en el rango dado.
- is not between
 - Filtrar donde el valor para el campo no está en el rango dado.
- exists
 - Filtrar donde está presente cualquier valor para el campo.
- does not exist
 - Filtro donde no hay valor presente para el campo.

Add filter



Filter

Edit Query DSL

host.architecture

is

x86_64

Label

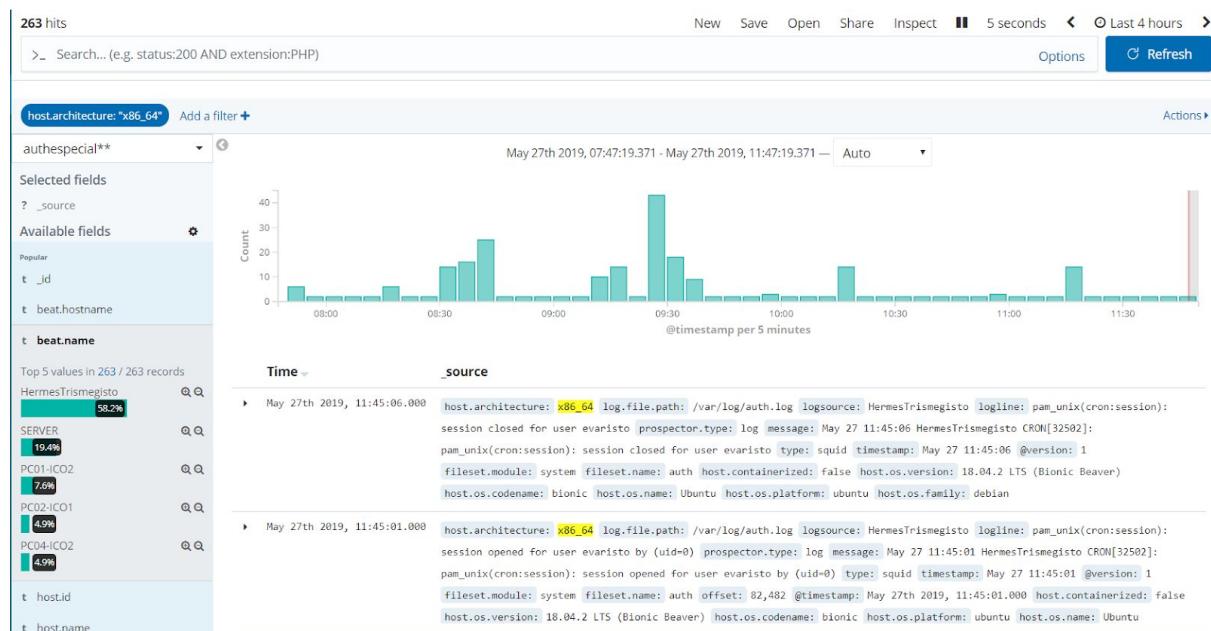
Optional

Cancel

Save

(Opcional) Especificamos una etiqueta para el filtro. Si se especifica una etiqueta, se mostrará debajo de la barra de consulta en lugar de la definición del filtro.

Hacemos clic en Guardar. El filtro se aplicará en la búsqueda y se mostrará debajo de la barra de consulta.



Administrar filtros

Para modificar un filtro, nos desplazamos sobre él y hacemos clic en uno de los botones de acción.



Habilitar filtro

Deshabilita el filtro sin quitarlo. Hacemos clic de nuevo para volver a habilitar el filtro. Las rayas diagonales indican que un filtro está deshabilitado.

Filtro de pin

Pin el filtro. Los filtros anclados persisten cuando cambias de contexto en Kibana. Por ejemplo, podemos fijar un filtro en Descubrir y permanecerá en su lugar cuando cambiemos a Visualizar. Hay que tener en cuenta que un filtro se basa en un campo de índice particular: si los índices que se buscan no contienen el campo en un filtro anclado, no tiene ningún efecto.

Invertir filtro

Cambia de un filtro positivo a un filtro negativo y viceversa.

Quitar filtro

Retira el filtro.



Editar filtro

Edita la definición del filtro. Permite actualizar manualmente el filtro y especificar una etiqueta para el filtro.

Para aplicar una acción de filtro a todos los filtros aplicados, haremos clic en Acciones y seleccionaremos la acción.

Visualizar datos del documento

Cuando se envía una consulta de búsqueda, los 500 documentos más recientes que coinciden con la consulta se enumeran en la tabla Documentos. Podemos configurar el número de documentos que se muestran en la tabla configurando la "discover:sampleSize" propiedad en Configuración avanzada.

Para ver los datos de campo de un documento, haremos clic en el botón Expandir a la izquierda de la entrada de la tabla del documento.

Para ver el documento JSON original, haremos clic en la pestaña JSON .

Para ver los datos del documento como una página separada, haremos clic en el enlace Ver documento único. Se puede marcar y compartir este enlace para proporcionar acceso directo a un documento en particular.

Añadir columnas de campo de la tabla de documentos

Para mostrar u ocultar la columna de un campo en la tabla Documentos, haremos clic en el botón Alternar columna en la tabla .

Para contraer los detalles del documento, haremos clic en el botón Contraer .

```
May 27th 2019, 12:15:05.000 log.file.path: /var/log/auth.log logsource: HermesTrismegisto logline: pam_unix(cron:session): session closed for user evaristo prospector.type: log message: May 27 12:15:05 HermesTrismegisto CRON[1553]: pam_unix(cron:session): session closed for user evaristo type: squid timestamp: May 27 12:15:05 @version: 1 fileset.module: system fileset.name: auth host.id: 96b6c9dfb2c6425291ba0d78d77bb108 host.containerized: false host.os.version: 18.04.2 LTS (Bionic Beaver) host.os.codename: bionic host.os.platform: ubuntu host.os.name: Ubuntu host.os.family: debian host.name: HermesTrismegisto
```

Table	JSON	View surrounding documents	View single document
⌚ @timestamp	⌚ ⚡ May 27th 2019, 12:15:05.000		
t @version	⌚ ⚡ 1		
t _id	⌚ ⚡ RqbJ-GoBPwMrGmABdqD_		
t _index	⌚ ⚡ authespecial		

Las columnas de campo agregadas reemplazan la `_sourcecolumn` en la tabla Documentos. Los campos agregados también se agregan a la lista Campos seleccionados .

Para reorganizar las columnas de campo, coloque el cursor sobre el encabezado de la columna que desea mover y haremos clic en el botón Mover a la izquierda o Mover a la derecha

Time ▾

- ▶ May 27th 2019, 12:17:01.000
- ▼ May 27th 2019, 12:17:01.000

	beat.hostname	host.architecture
	HermesTrismegisto	x86_64
	PC01-ICO2	x86_64

Table JSON View surrounding documents View single document

```

{
  "@timestamp": "May 27th 2019, 12:17:01.000",
  "@version": "1",
  "_id": "wqbL-GoBPwMrGmABJ-w6",
  "_index": "authespécial",
  "_score": "-",
  "_type": "doc",
  "beat": {
    "hostname": "PC01-ICO2",
    "name": "PC01-ICO2"
  },
  "beat.version": "6.7.1",
  "event": {
    "dataset": "system.auth"
  },
  "fileset": {
    "module": "system",
    "name": "auth"
  },
  "host": {
    "architecture": "x86_64"
  },
  "host.containerized": false,
  "host.id": "c97c2fbc390409d22b85f4895c9a7977"
}

```

Eliminar columnas de campo de la tabla de documentos

Para eliminar una columna de campo de la tabla Documentos, coloque el cursor sobre el encabezado de la columna que desea eliminar y haremos clic en el botón Eliminar ✕:

beat.hostname ✕ »

Remove column

Visualize

Visualizar nos permite crear visualizaciones de los datos de los índices de Elasticsearch.

Las visualizaciones de Kibana se basan en consultas de Elasticsearch. Al utilizar una serie de agregaciones de Elasticsearch para extraer y procesar los datos, podemos crear gráficos que nos muestren las tendencias, los picos y las caídas que necesitamos conocer.

Podemos crear visualizaciones a partir de una búsqueda guardada desde Discover o comenzar con una nueva consulta de búsqueda.

The screenshot shows the Kibana interface with the 'Visualize' tab selected in the sidebar. The main area displays a list of saved visualizations. At the top right of the list area is a search bar and a blue '+' button. Below the search bar, the text '1-20 of 375' is displayed. The list contains 20 items, each with a checkbox, a title, a type, and a small icon. The types listed are 'Data Table', 'Visual Builder', and 'Coordinate Map'. The titles include 'Accept / Recvfrom Unique Address Table [Auditbeat Auditd]', 'Accepts and Handled Rate [Metricbeat Nginx]', 'Accesos susuarios systemd y sshd', 'Access logs over time [Filebeat Nginx]', 'Access map [Filebeat IIS]', 'Access Map [Filebeat Nginx]', 'Access Map [Filebeat Nginx] [ML]', 'Access Map [Filebeat Traefik]', 'Access Map [Filebeat Traefik] [ML]', and 'Acciones Ejecutadas En Internet'.

Todas estas gráficas estarán disponibles para ser añadidas en los paneles. Si hacemos clic en el botón de añadir, podremos crear una nueva gráfica.

Para crear una visualización:

haremos clic en Visualizar en la navegación lateral.

haremos clic en el botón Crear nueva visualización o en el botón + .

Elegiremos el tipo de visualización:

This screenshot is identical to the one above it, showing the Kibana Visualize interface. However, the blue '+' button at the top right of the visualization list is now highlighted with a red box, indicating it is the target for creating a new visualization.

New Visualization

Filter

Area Controls Coordinate Map Data Table

Gauge Goal Heat Map Horizontal Bar

Line Markdown Metric Pie

Region Map Tag Cloud Timelion Vega

Vertical Bar Visual Builder

Tag Cloud

A group of words, sized according to their importance

Especificaremos una consulta de búsqueda para recuperar los datos para la visualización:

Para ingresar nuevos criterios de búsqueda, seleccionaremos el patrón de índice para los índices que contienen los datos que deseamos visualizar. Esto abre el generador de visualización con una consulta de comodín que coincide con todos los documentos en los índices seleccionados.

Para crear una visualización a partir de una búsqueda guardada, haremos clic en el nombre de la búsqueda guardada que deseamos usar. Esto abre el generador de visualización y carga la consulta seleccionada.

From a New Search, Select Index

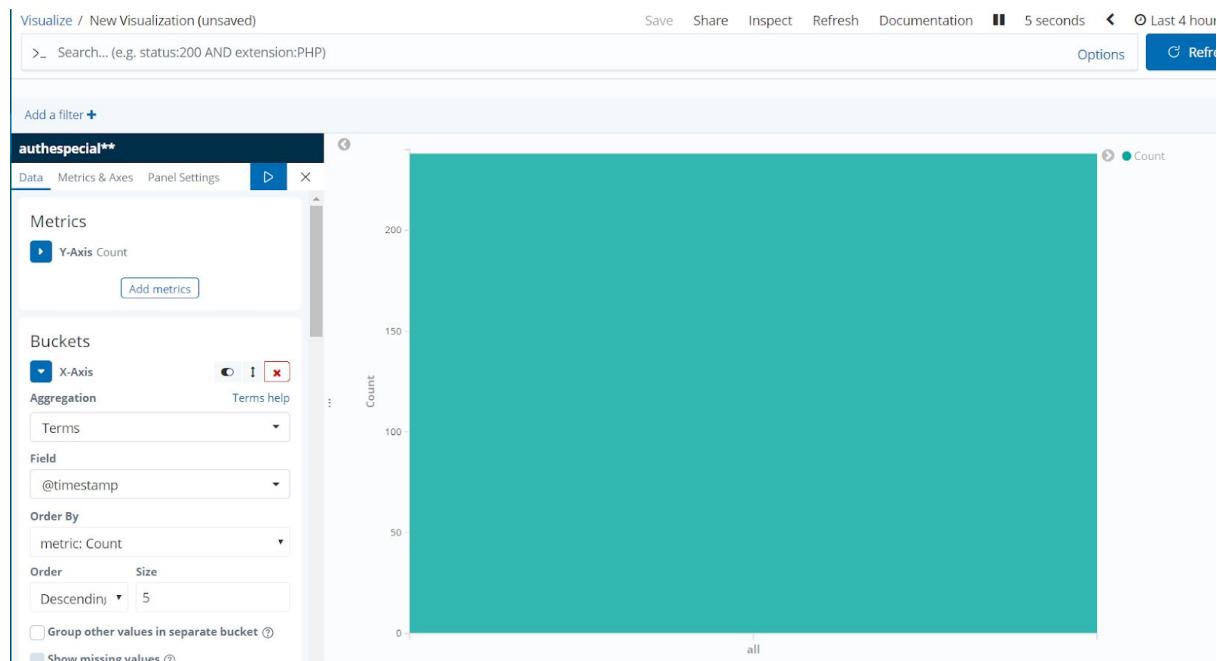
<input type="text"/> Filter...	10 of 10
Name ▲	
packetbeat-*	
squid-tikitikiv3	
authpersonalitydequenuevo*	
squid-tikitikiv3*	
auditbeat-*	
journalbeat-*	
metricbeat-*	
filebeat*	
filebeat-*	
authespecial**	

Or, From a Saved Search

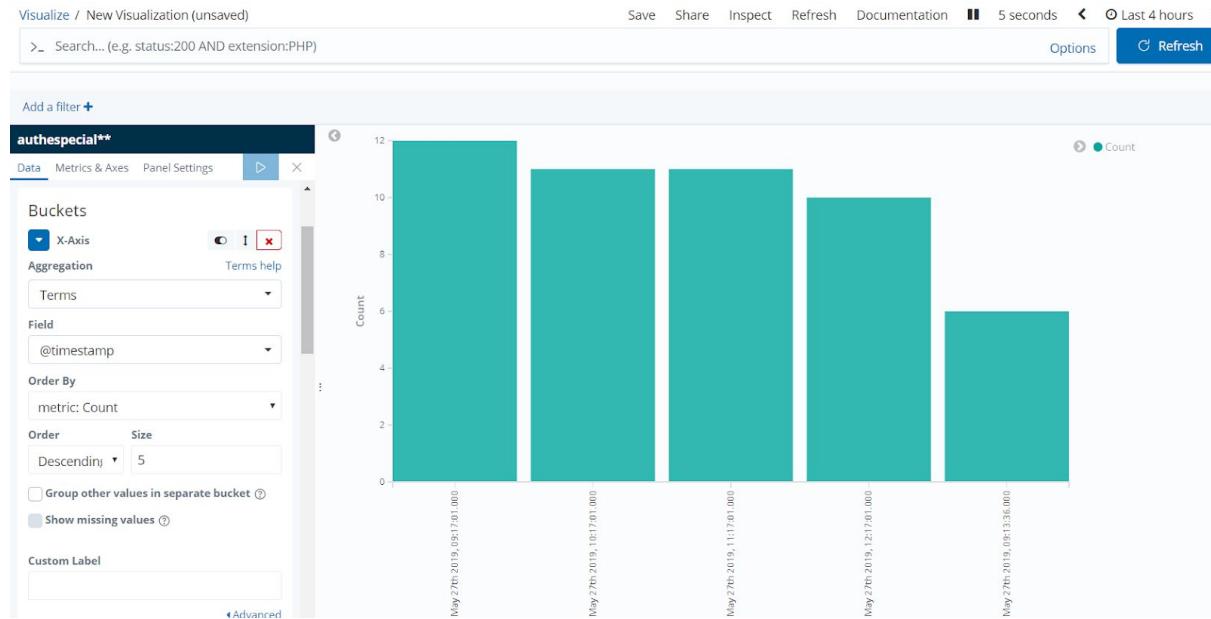
<input type="text"/> Saved Searches Filter...	1-20 of 88	Manage saved searches
Name ▲		
Accesos Usuarios		
Alerts [Suricata]		
All Logs [Filebeat PostgreSQL]		
All logs [Filebeat Kafka]		
All logs [Filebeat MongoDB]		
Apache HTTPD		
Apache access logs [Filebeat Apache2]		
Apache errors log [Filebeat Apache2]		
Audit Event Table [Auditbeat Auditd]		
Audit Events [Filebeat Auditd]		
Cache Transactions [Packetbeat]		
Cassandra QueryView		
DB transactions		
DFR packages installed [OSquery results]		

Vamos a intentar comprenderlo con un ejemplo

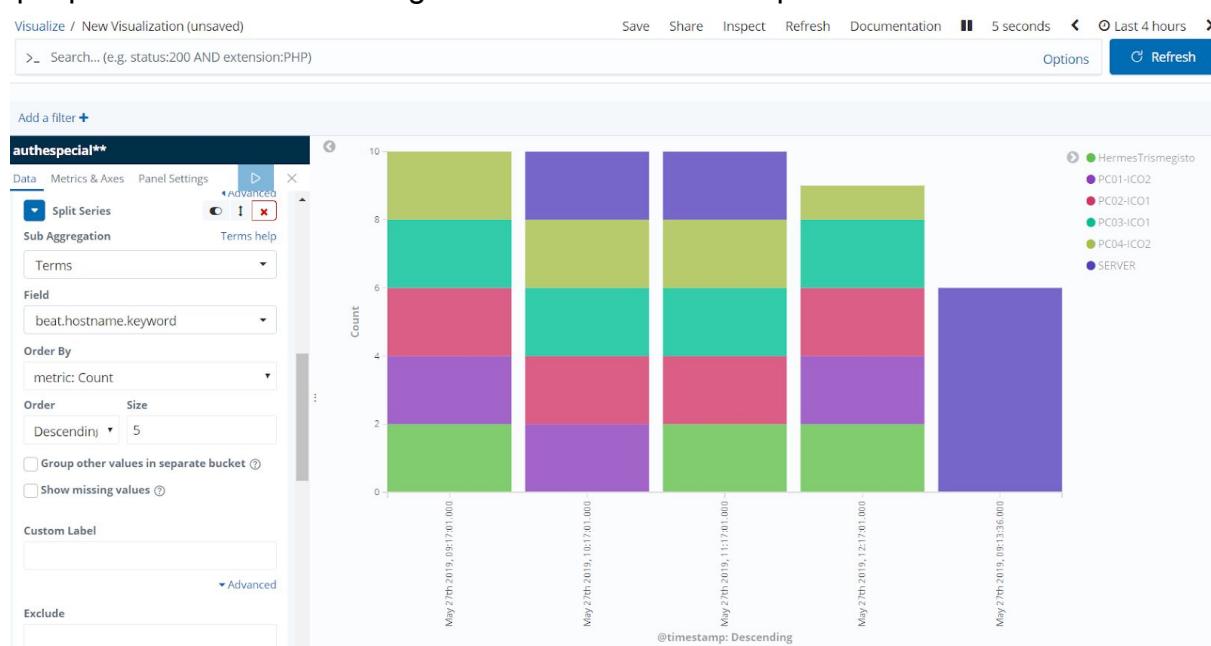
Indicamos “Count” en el eje Y, y nos muestra el número de logs de éste índice recibido.



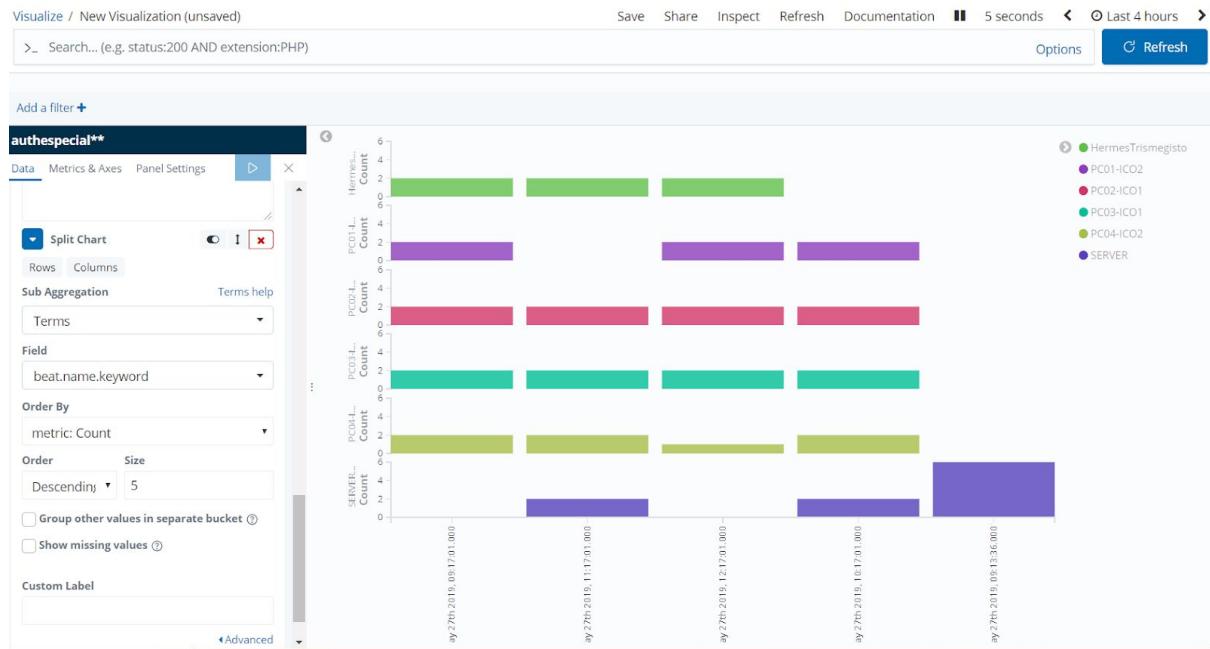
Ahora además, le indicamos que en el eje X nos divida por marca de tiempo, los eventos.



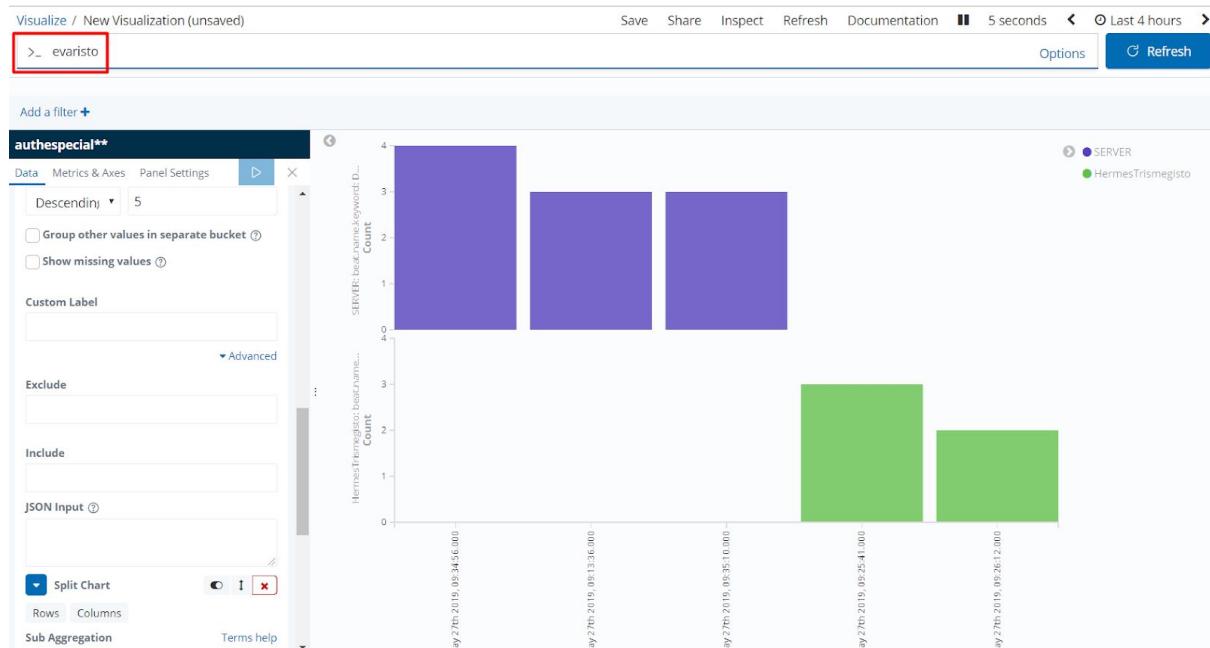
A continuación indicamos que nos divida las barras en función del nombre de host al que pertenezca el total de logs de cada marca de tiempo



Y por último indicamos que nos agrupe por nombre de host.



Ahora podemos afinar la búsqueda con la sintaxis de Kibana



Ya tenemos filtrado solo los mensajes de log correspondiente al usuario “evaristo”
 Además vamos a filtrar, solo por los mensajes del programa “sshd”
 Podríamos hacerlo, mediante la misma consulta (con operadores AND u OR) o con un filtro como veremos a continuación:

Add a filter +

Add filter

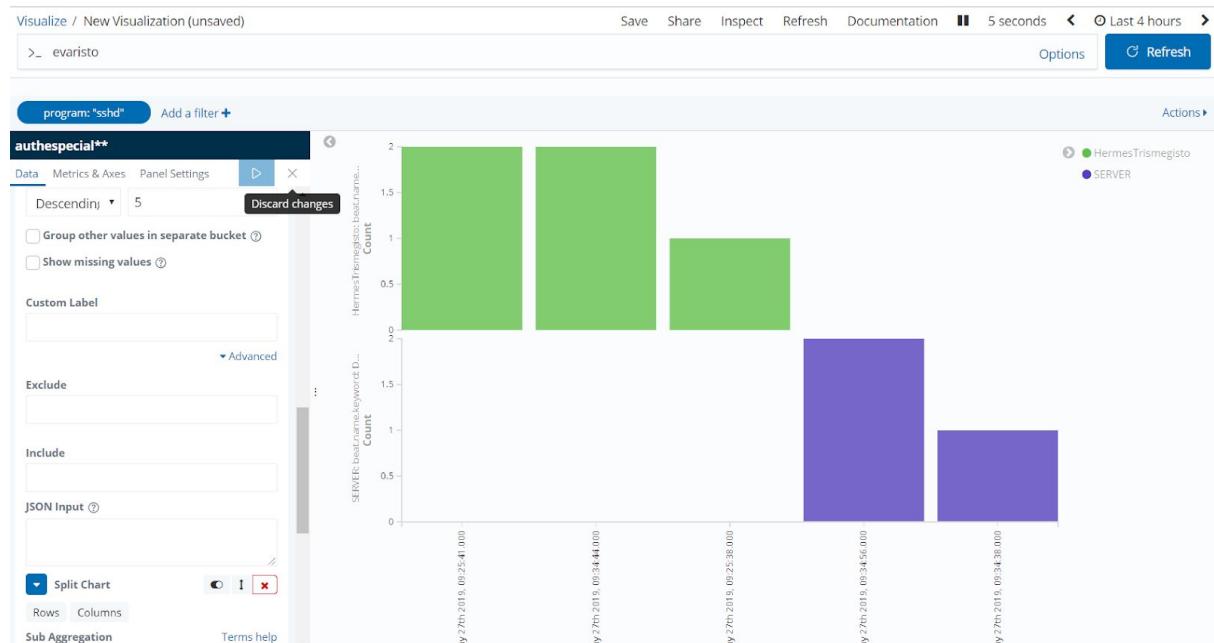
Filter

program is sshd

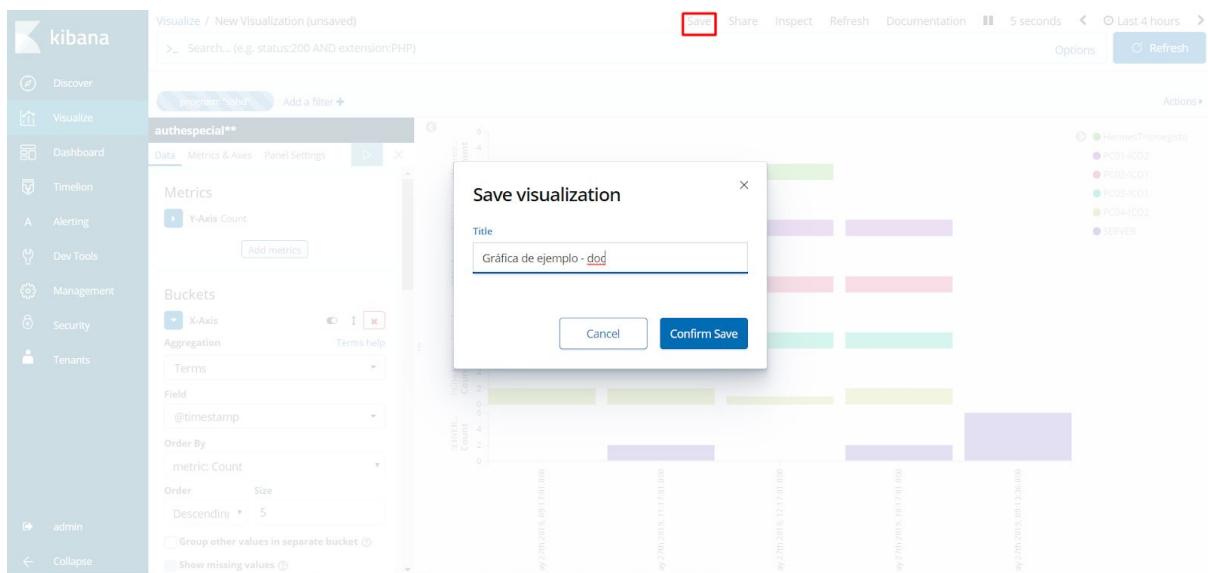
Label

Optional

Cancel Save



Una vez estemos conforme con nuestra gráfica, la guardaremos:

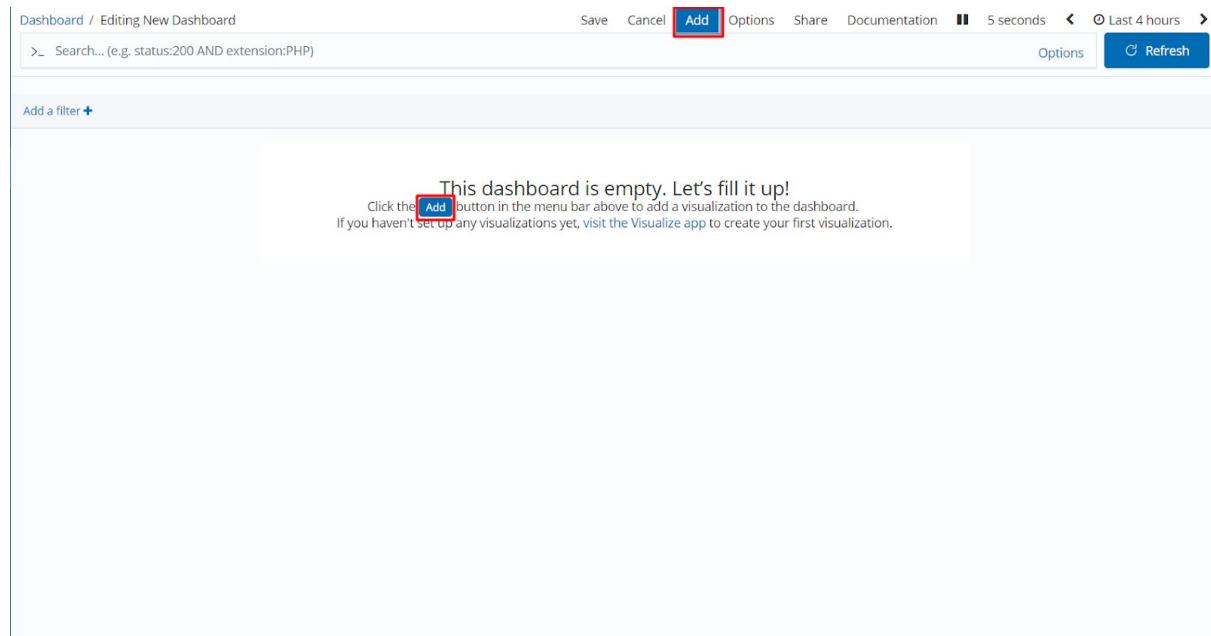


Desde este momento, podremos volver a abrirla para consultarla o editarla, y lo más interesante, podremos añadirla a un panel.

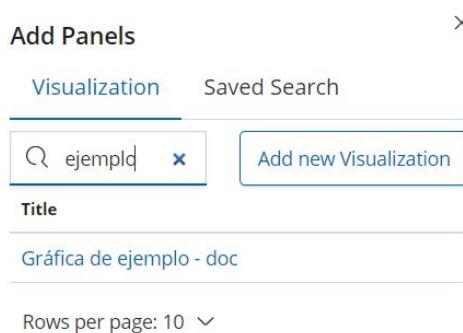
Dashboards

Title	Description	Actions
[Auditbeat Auditd] Executions	Summary of Linux kernel audit events.	Edit
[Auditbeat Auditd] Overview	Summary of socket related syscall events.	Edit
[Auditbeat Auditd] Sockets	Monitor file integrity events.	Edit
[Auditbeat File Integrity] Overview	Filebeat Apache2 module dashboard	Edit
[Filebeat Apache2] Access and error logs	Dashboard for the Auditbeat Filebeat module	Edit
[Filebeat Auditd] Audit Events	Filebeat HAProxy module dashboard	Edit
[Filebeat HAProxy] Overview	Filebeat Icinga module dashboard for the debug logs	Edit
[Filebeat Icinga] Debug Log	Filebeat Icinga module dashboard for the main log files	Edit
[Filebeat Icinga] Main Log	Filebeat Icinga module dashboard for startup errors	Edit
[Filebeat Icinga] Startup Errors	Dashboard for the Filebeat IIS module	Edit
[Filebeat IIS] Access and error logs	Filebeat Kafka module dashboard	Edit
[Filebeat Kafka] Overview		

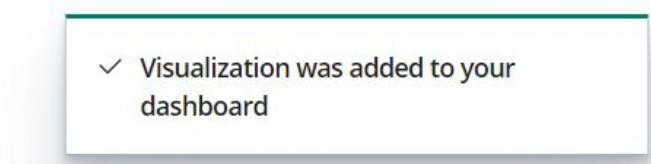
Para añadir un nuevo tablero hacemos clic en “Create new dashboard”



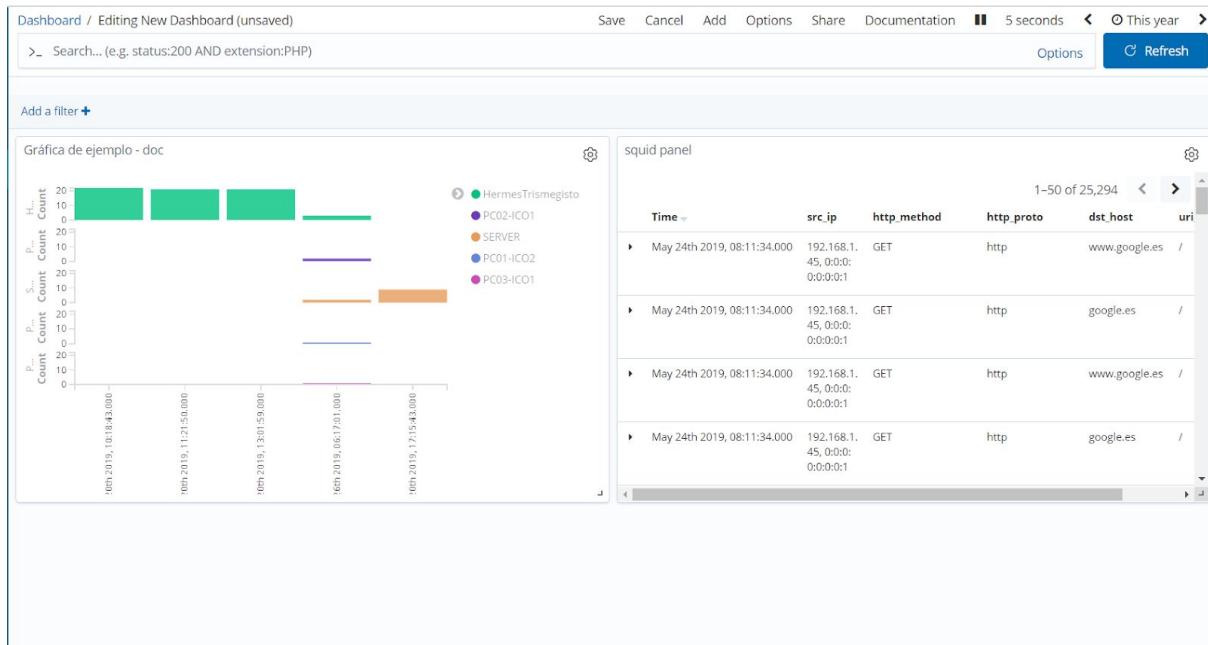
Nos aparecerá por defecto un tablero vacío, ahora le daremos a “Add” para añadir una primera gráfica:



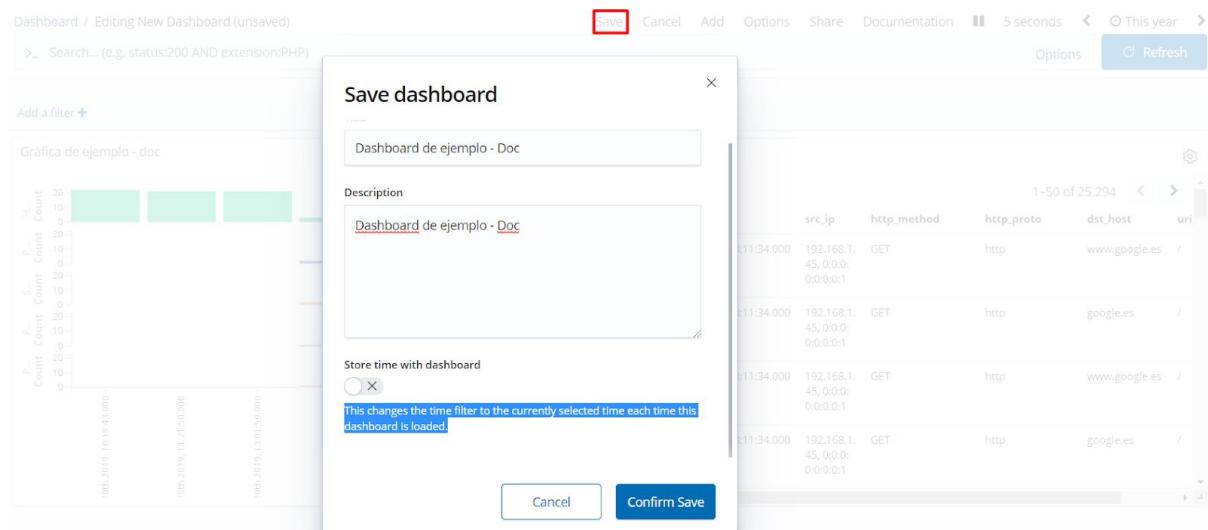
Podemos añadir tanto gráficas como consultas que hayamos guardado. Simplemente hacemos clic sobre el objeto deseado.



Podemos añadir tantas gráficas y/o consultas guardadas como necesitemos.



Una vez hayamos terminado, le daremos a “Guardar”

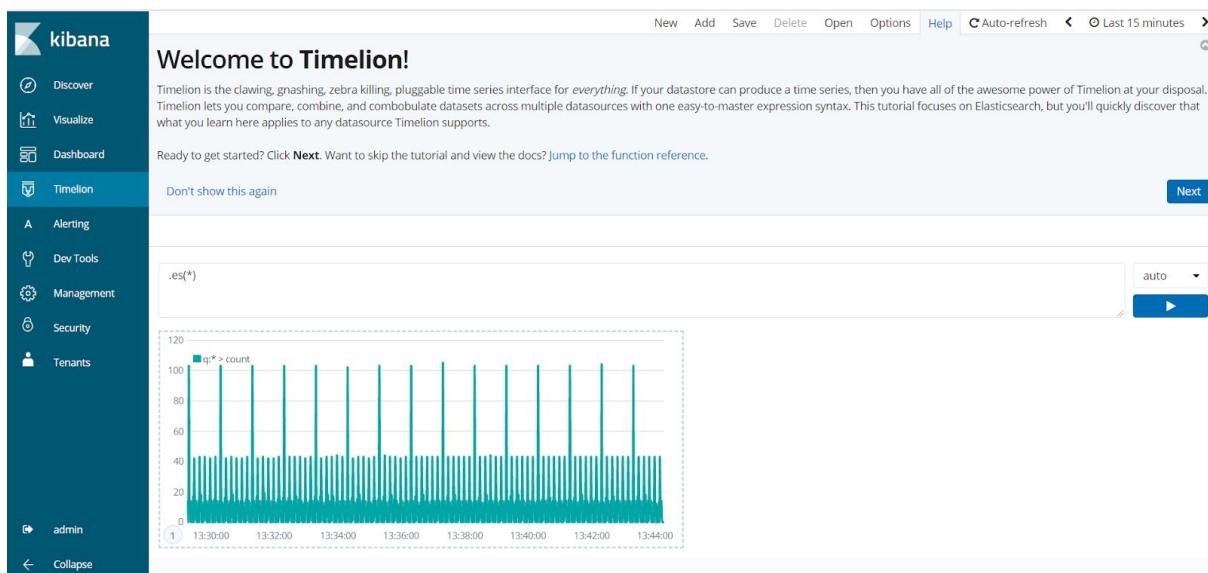


Podemos aplicar filtros y búsquedas nuevas sobre el panel, si estamos en modo edición podremos guardar el panel de nuevo o crear un nuevo panel a partir del anterior. Si no estamos en modo edición, podremos aplicar filtros pero no persistentes.

Timelion

Timelion es una herramienta bastante poderosa para analizar datos de series de tiempo y complementa las capacidades tradicionales de visualización de Kibana. Permite ejecutar cálculos matemáticos avanzados, como dividir y restar métricas, calcular derivadas y promedios móviles, y por supuesto visualizar los resultados de estos cálculos.

Timelion todavía está "en desarrollo" y la documentación aún es un poco incompleta, por lo que se puede esperar una sinuosa curva de aprendizaje (por ejemplo, la documentación de la función carece de ejemplos).



Alerting

En esta sección podremos configurar las alertas ante ciertos eventos.

The screenshot shows the Kibana Alerting Dashboard. The left sidebar has a dark blue background with white icons and labels: Discover, Visualize, Dashboard, Timeline, Alerting (which is highlighted in blue), Dev Tools, Management, Security, and Tenants. The top navigation bar has tabs for Alerting and Dashboard, with Dashboard being the active tab. Below the navigation is a search bar with placeholder text "Search" and a dropdown menu for "All severity levels". A button labeled "Acknowledge" is in the top right corner. The main content area is titled "Alerts" and contains a table with one row of data. The table columns are: Alert start time (sorted), Alert end time, Monitor name, Trigger name, Severity, State, and Time acknowledged. The data row shows: 05/11/19 2:30 pm, -, test, evtrompa@gmail.com, 1, Deleted, -. Below the table is a dropdown menu for "Rows per page: 20".

Alert start time ↓	Alert end time	Monitor name	Trigger name	Severity	State	Time acknowledged
05/11/19 2:30 pm	-	test	evtrompa@gmail.com	1	Deleted	-

Dev Tools

En esta sección podremos interactuar con la apiREST.

The screenshot shows the Kibana Dev Tools interface. On the left, a sidebar lists navigation options: Discover, Visualize, Dashboard, Timeline, Alerting, Dev Tools (which is selected), Management, Security, and Tenants. At the bottom of the sidebar are links for 'admin' and 'Collapse'. The main area is titled 'Dev Tools' and 'Console'. In the 'Console' tab, there are two code editors. The left editor contains a single-line Elasticsearch search query: `GET _search`. The right editor displays the JSON response from the search query, which includes metrics like 'took', 'shards', and 'hits', along with detailed information for each hit such as '_index', '_type', '_id', '_score', and '_source'.

```
1 GET _search
2 {
3   "query": {
4     "match_all": {}
5   }
6 }
```

```
1 {
2   "took": 725,
3   "timed_out": false,
4   "num_reduce_phases": 3,
5   "shards": {
6     "total": 1100,
7     "successful": 1100,
8     "skipped": 0,
9     "failed": 0
10 },
11 },
12 "hits": {
13   "total": 26439461,
14   "max_score": 1.0,
15   "hits": [
16     {
17       "_index": ".kibana_152937574_admintenant_1",
18       "_type": "doc",
19       "_id": "config6.7.1",
20       "_score": 1.0,
21       "_source": {
22         "config": {
23           "buildnum": 20266
24         },
25         "type": "config",
26         "updated_at": "2019-05-21T07:54:16.681Z"
27       }
28     },
29     {
30       "_index": ".kibana_152937574_admintenant_2",
31       "_type": "doc",
32       "_id": "config6.7.1",
33       "_score": 1.0,
34       "_source": {
35         "config": {
36           "buildnum": 20266
37         },
38         "type": "config",
39         "updated_at": "2019-05-21T07:54:16.681Z"
40       }
41     }
42   ]
43 }
```

Management

Ya hemos visto parte de esta sección, cuando configuramos los patrones de índice.

Index Patterns

Name	Type	Format	Searchable	Aggregatable	Excluded
@timestamp	date		●	●	<input type="checkbox"/>
@version	string		●		<input type="checkbox"/>
@version.keyword	string		●	●	<input type="checkbox"/>
_id	string		●	●	<input type="checkbox"/>
_index	string		●	●	<input type="checkbox"/>
_score	number				<input type="checkbox"/>
_source	_source				<input type="checkbox"/>
_type	string		●	●	<input type="checkbox"/>
beat.hostname	string		●		<input type="checkbox"/>
beat.hostname.keyword	string		●	●	<input type="checkbox"/>

Además podemos exportar e importar objetos:

Saved Objects

Type	Title	Actions
Icon	[Metricbeat HAProxy] HTTP frontend	<input type="checkbox"/>
Icon	[Metricbeat HAProxy] Backend	<input type="checkbox"/>
Icon	[Metricbeat Kubernetes] Overview	<input type="checkbox"/>
Icon	[Metricbeat System] Host overview	<input type="checkbox"/>
Icon	[Metricbeat Kafka] Overview	<input type="checkbox"/>
Icon	[Packetbeat] PgSQL performance	<input type="checkbox"/>
Icon	[Packetbeat] Thrift performance	<input type="checkbox"/>

Así como tocar configuraciones avanzadas de Kibana.

Advanced Settings

⚠ Caution: You can break stuff here
Be careful in here, these settings are for very advanced users only. Tweaks you make here can break large portions of Kibana. Some of these settings may be undocumented, unsupported or experimental. If a field has a default value, blanking the field will reset it to its default which may be unacceptable given other configuration directives. Deleting a custom setting will permanently remove it from Kibana's config.

General	
Quote CSV values	csv:quoteValues Should values be quoted in csv exports? <input checked="" type="checkbox"/> On
CSV separator	csvseparator Separate exported values with this string <input type="text" value=","/>
Date format	dateFormat When displaying a pretty formatted date, use this format <input type="text" value="MMMM Do YYYY, HH:mm:ss.SSS"/>
Day of week	dateFormat:dow What day should weeks start on? <input type="text" value="Sunday"/>

Security

En esta sección podremos configurar todo lo relativo a la seguridad de Kibana.

The screenshot shows the Kibana Security interface. On the left is a sidebar with icons for Discover, Visualize, Dashboard, Timelion, Alerting, Dev Tools, Management, Security (which is selected), and Tenants. Below the sidebar are buttons for admin and Collapse. The main area is titled "Security" and contains three sections: "Permissions and Roles" with icons for Role Mappings, Roles, and Action Groups; "Authentication Backends" with an icon for Internal User Database; and "System" with icons for Authentication & Authorization and Purge Cache.

Tenants

En esta sección podremos crear distintos “entornos de uso” separados en Kibana, con el fin de compartir el uso de Kibana con más departamentos.

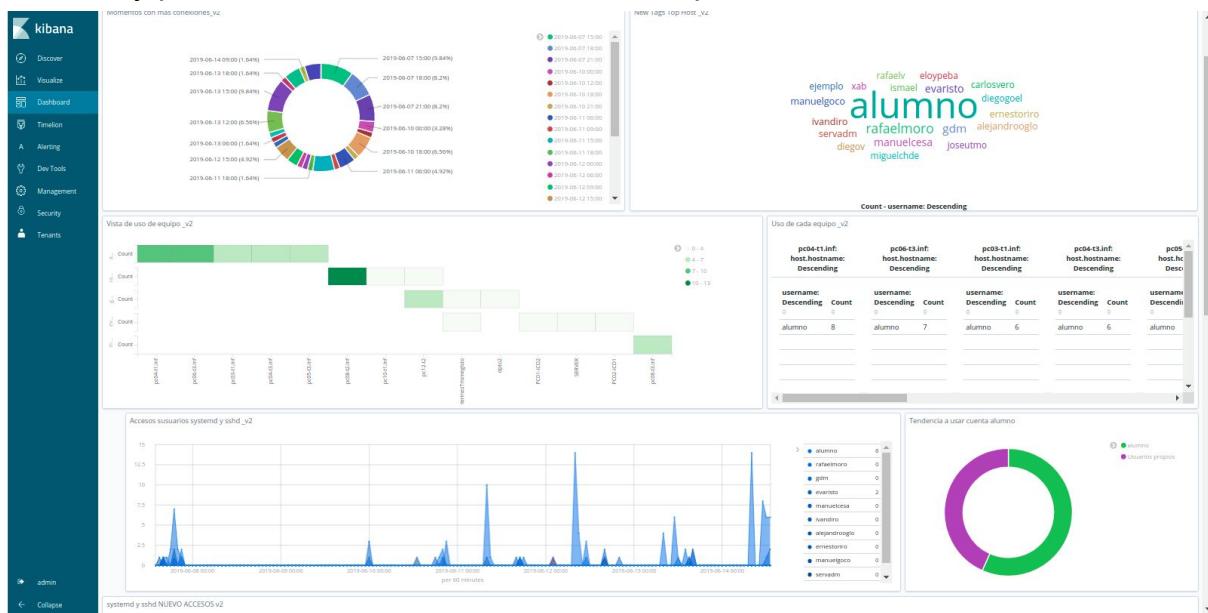
The screenshot shows the Kibana Tenant selection interface. The sidebar is identical to the previous one. The main area is titled "Select Tenant" and shows a table of tenants. The table has columns for Name and Permissions. It lists "Global" (read/write), "Private" (read/write), and "admin_tenant" (read/write). For each tenant, there are "Show Dashboards" and "Show Visualizations" buttons, and a "Select" button. The "Private" tenant is highlighted. At the top right of the table, it says "Active tenant: Private".

Solución a medida del departamento

Con el conocimiento expuesto en los puntos anteriores, se ha diseñado una solución de paneles muy concreta para las necesidades del departamento de informática.

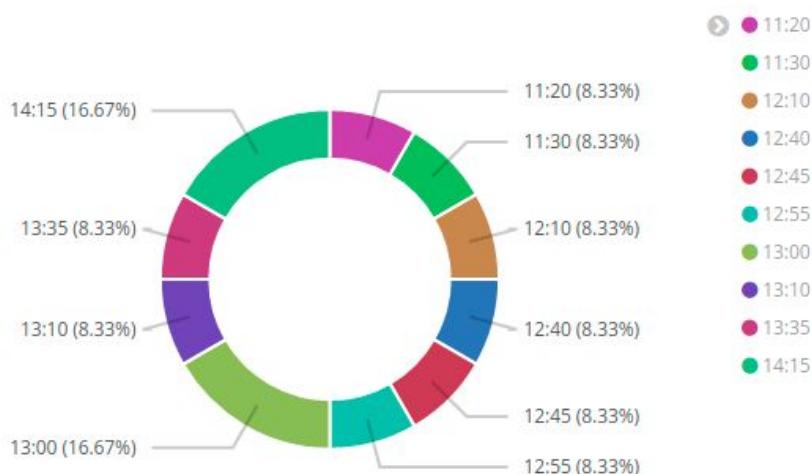
Accesos y navegación de usuarios

Se ha integrado en un panel, la información relevante en cuanto accesos de alumnos y profesores en las instalaciones del departamento.

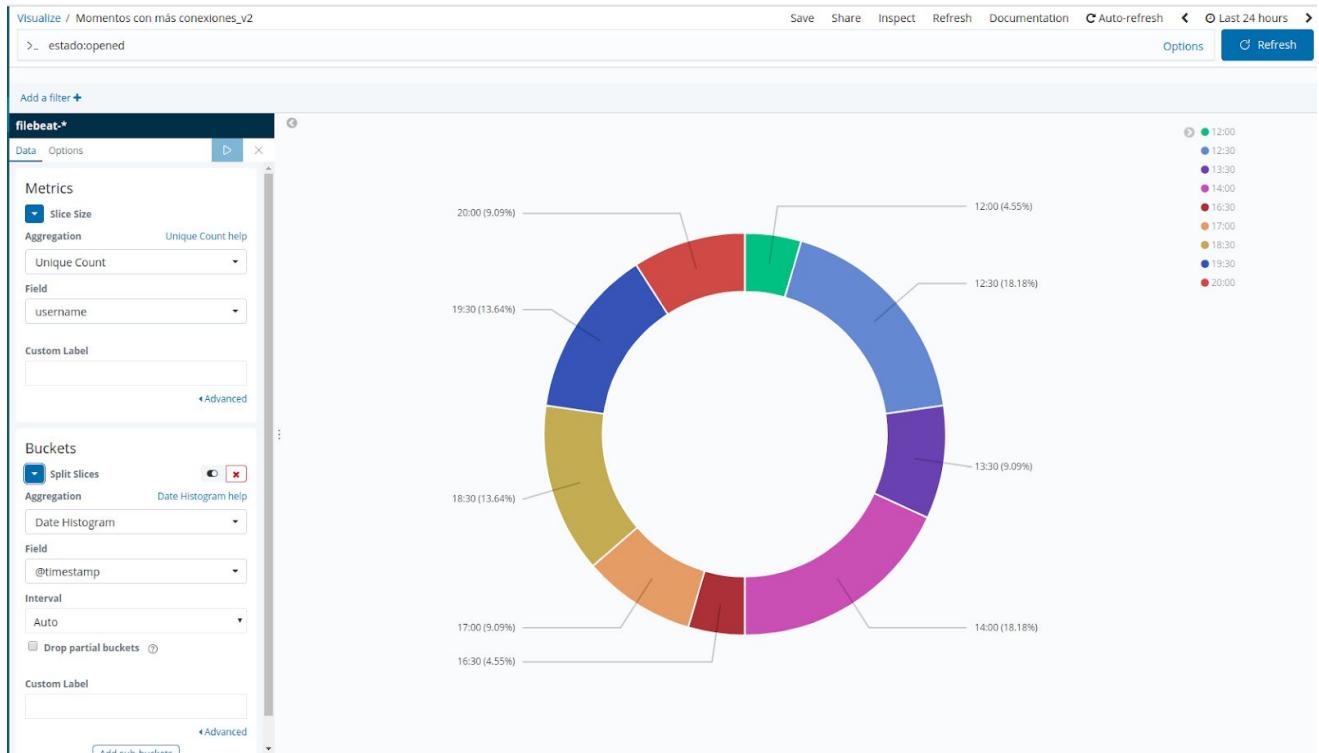


En la siguiente gráfica observamos las horas con más inicios de sesión.

Momentos con más conexiones_v2



Esta gráfica se consigue de la siguiente manera:

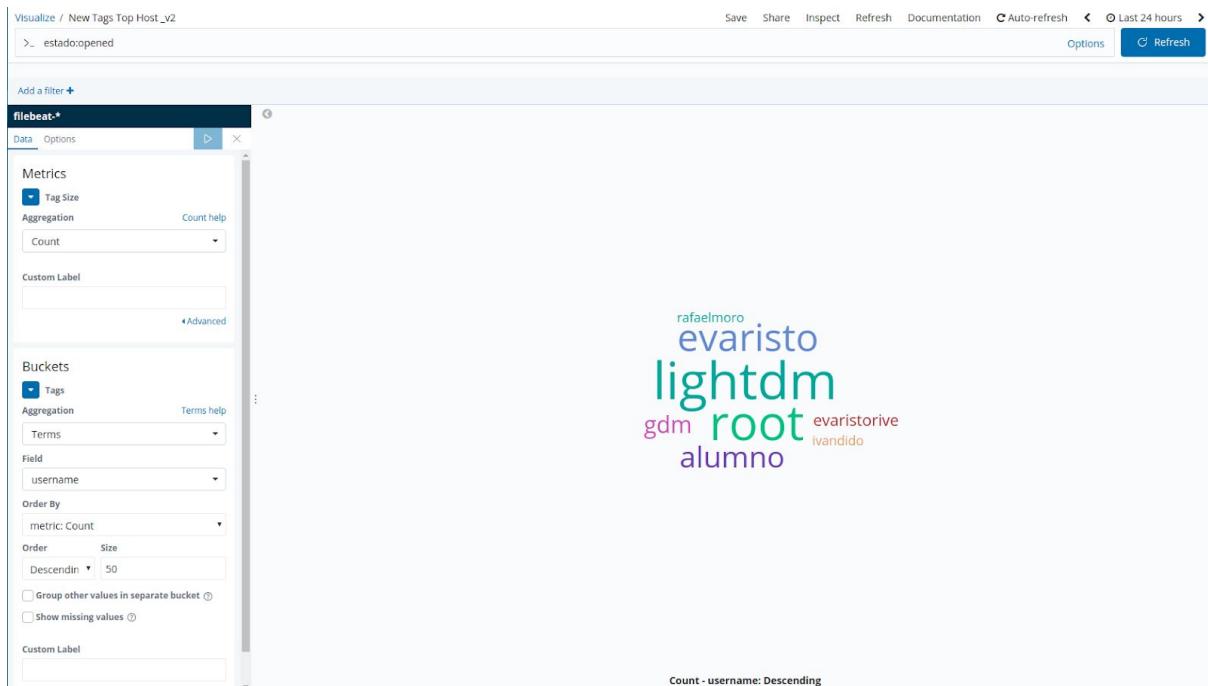


Variará según el rango de tiempo que seleccionemos, pudiendo ir hacia atrás (nos mostraría días o hacia delante (nos mostrarían minutos concretos)

Acompañándola vemos unos tags en forma de nube, que nos muestran los nombres de usuario que más actividad tienen. Podemos confirmar que se abusa del usuario "Alumno" (Podría denotar una cierta tendencia a olvidar la contraseña del usuario personal)



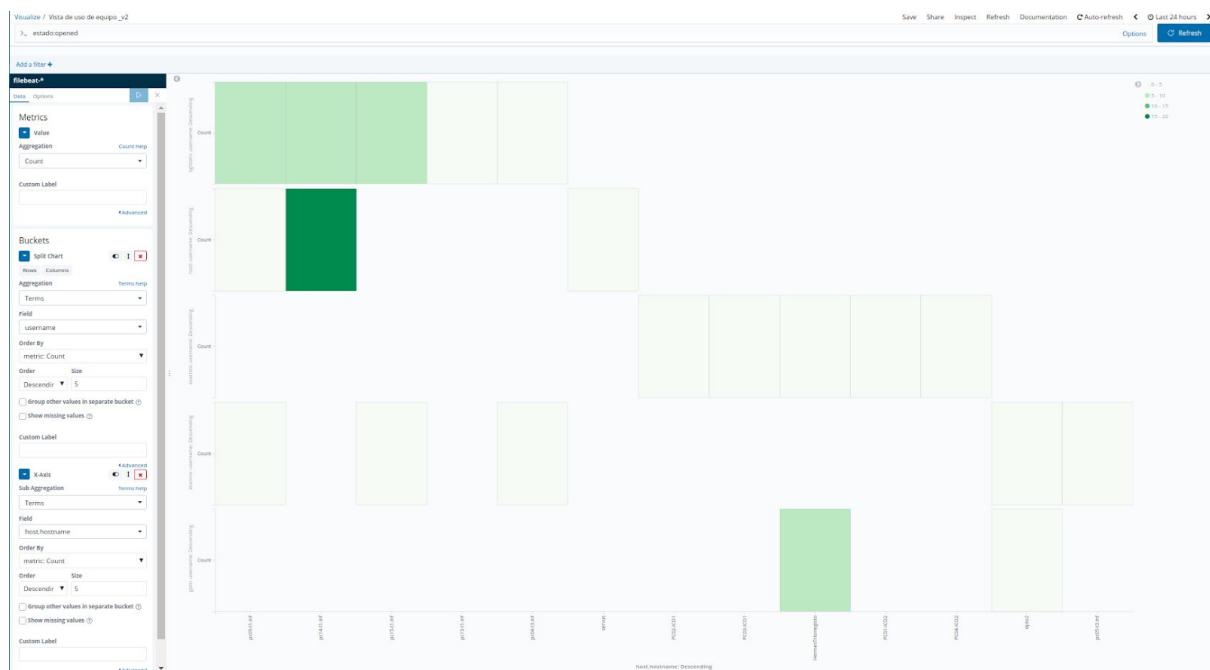
La nube de tags está construída de la siguiente manera:



Más abajo vemos el inventario de equipos, donde podemos ver por cada equipo, qué alumnos o profesores lo han estado utilizando, para el tiempo que indiquemos.



El gráfico de la izquierda está montado de la siguiente manera:

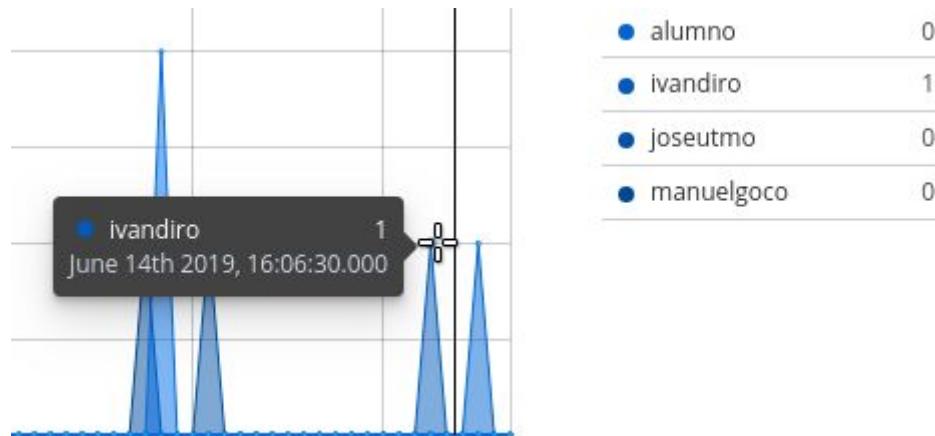


Mientras que la tabla de la derecha está montada así:

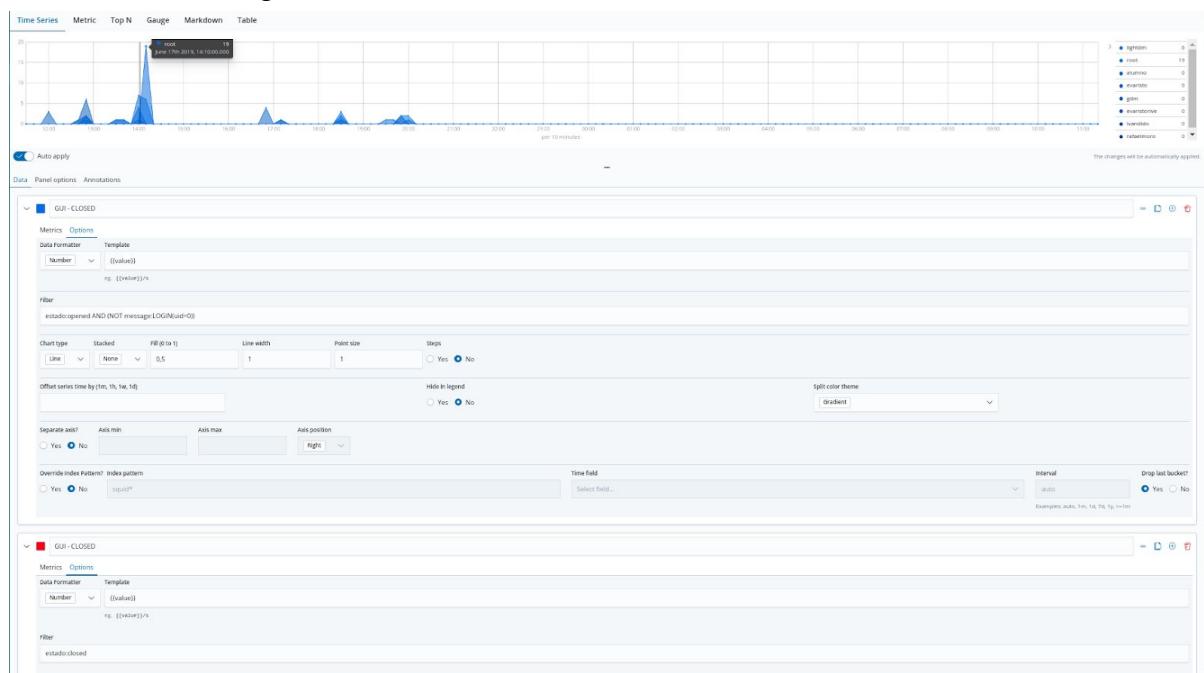
The visualization is a table titled "Visualize / Uso de cada equipo _v2". It displays data across multiple rows and columns. The columns represent different hosts and their usage counts. The table includes a header row with column names like "PC02-ICO1: host.hostname: Descending" and "username: Descending Count". The data rows show counts for various users across different hosts. The left sidebar contains sections for Metrics (Metric, Aggregation, Count), Buckets (Split Rows, Terms, host.hostname), and Custom Label. The top header includes Save, Share, Inspect, Refresh, Documentation, Auto-refresh, Last 24 hours, Options, and Refresh buttons. The table has export options for raw and formatted data at the bottom.

PC02-ICO1: host.hostname: Descending	PC03-ICO1: host.hostname: Descending	HermesTrismegisto: host.hostname: Descending	PC01-ICO2: host.hostname: Descending	PC04-ICO2: host.hostname: Descending	pc15-t1.inf: host.hostname: Descending	pc04-t1.inf: host.hostname: Descending	pc09-t1.inf: host.hostname: Descending	pc05-t3.inf: host.hostname: Descending	pc1 host.J. Descending
username: Descending Count	username: Descending Count	username: Descending Count	username: Descending Count	username: Descending Count	username: Descending Count	username: Descending Count	username: Descending Count	username: Descending Count	username: Descending Count
evaristo 4	evaristo 4	evaristo 2	evaristo 2	evaristo 2	alumno 3	alumno 2	alumno 1	alumno 1	ivanidic 1
	gdm 6								

A modo esquemático vemos una gráfica temporal, que nos sitúa en el tiempo, los inicios de sesión de alumnos, y profesores.



Realizada de la siguiente manera:

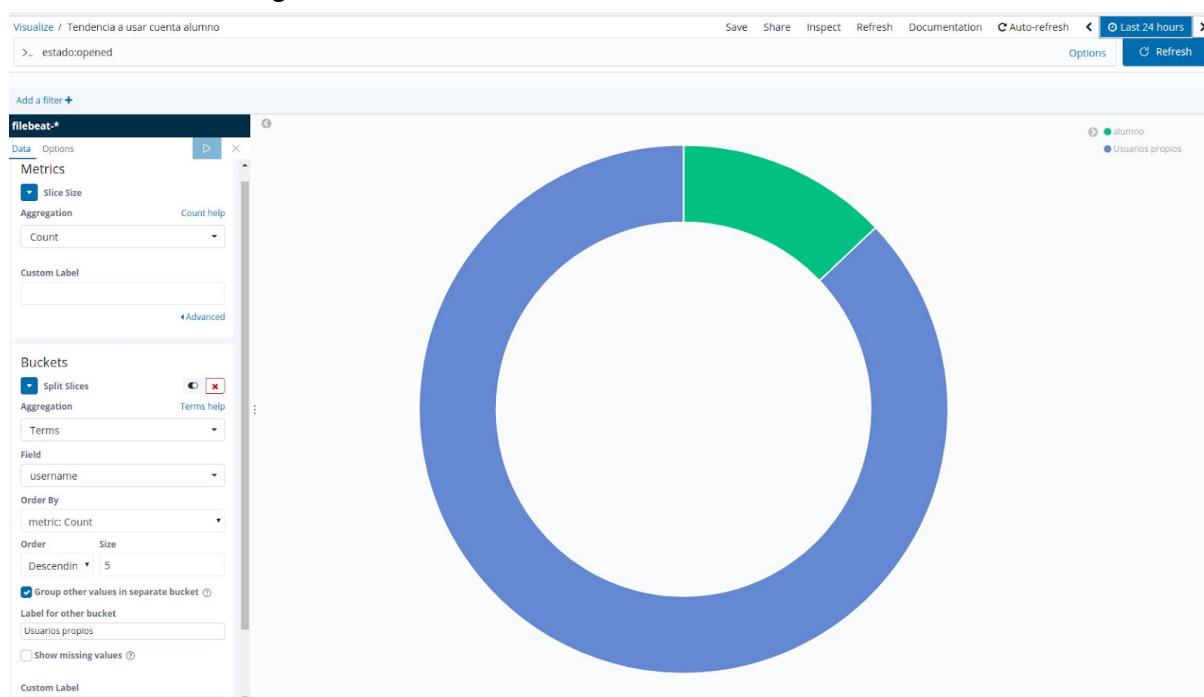


Importante filtrar por “estado:opened AND (NOT message:LOGIN(uid=0))” para no tener los accesos por duplicado, ya que en el caso de los accesos mediante Login, (tty1 por ejemplo) suele ir de la mano los dos tipos de mensajes en el auth.log

A la derecha de dicha gráfica, observamos una rueda que marca la tendencia a usar la cuenta alumno, por cada rango de tiempo que indiquemos.



Construida de la siguiente manera:

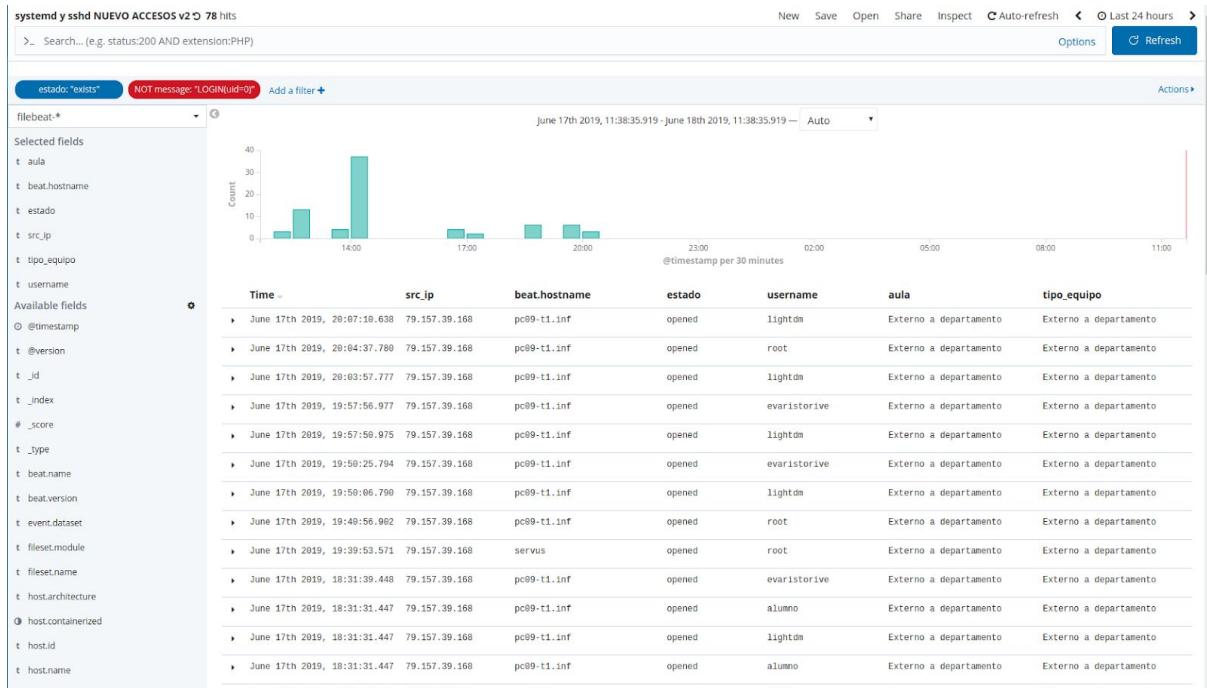


Y para terminar esta sección de accesos, observamos la vista de log, de todos los registros.

Time	src_ip	beat.hostname	estado	username
▶ June 14th 2019, 16:07:04.940	79.157.39.168	pc06-t3.inf	opened	alumno
▶ June 14th 2019, 16:06:33.332	79.157.39.168	pc11-t3.inf	opened	ivandiro
▶ June 14th 2019, 16:04:14.930	79.157.39.168	pc06-t3.inf	opened	joseutmo

opened	evaristo	ICO3	PC de aula
opened	evaristo	ICO3	PC de aula

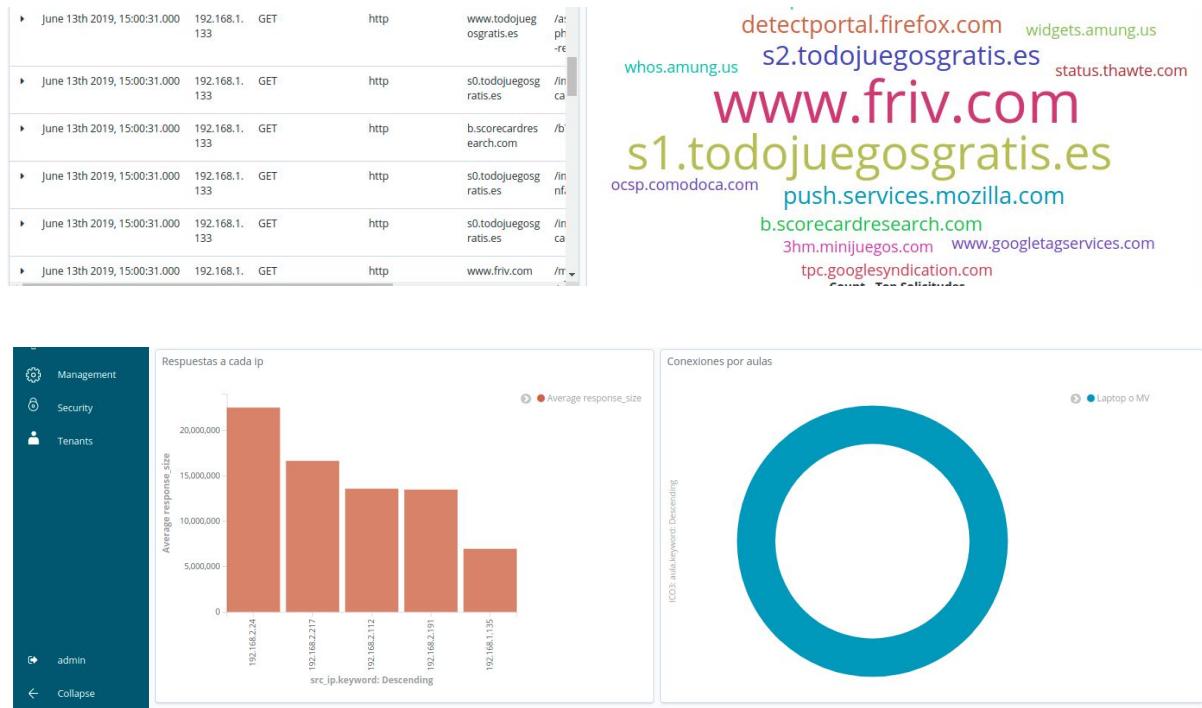
Dicha vista de log está basada en una búsqueda guardada, y es la siguiente:



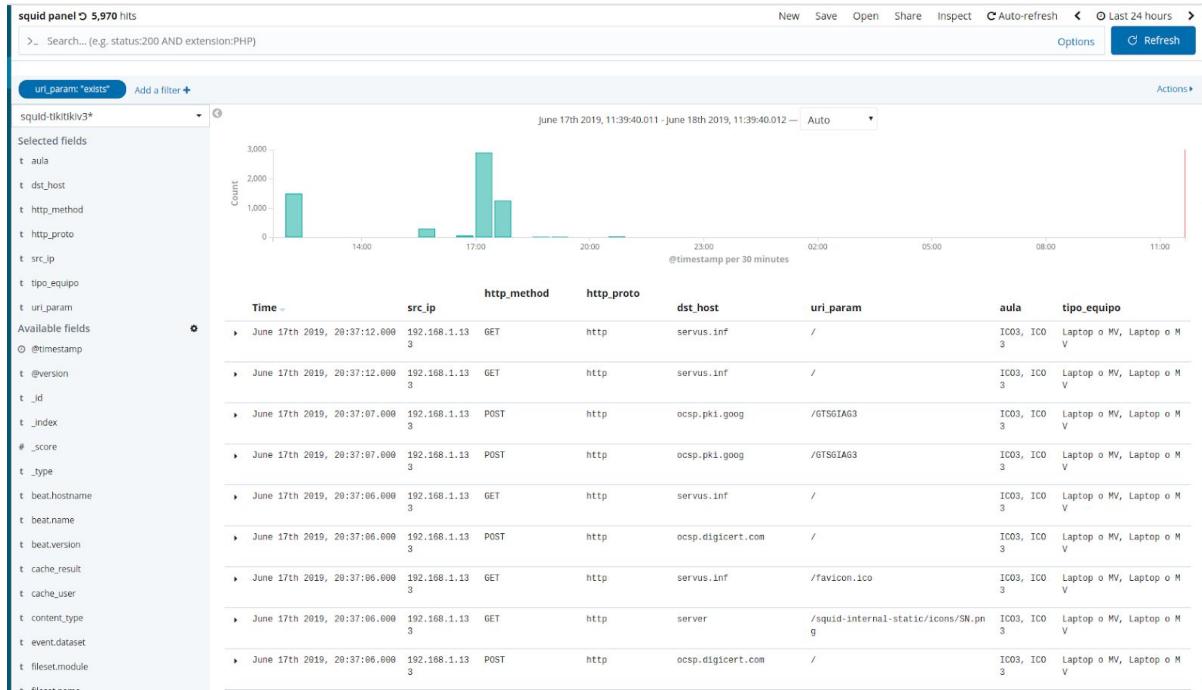
En la parte inferior del panel, nos da información acerca de los sitios por los que han estado navegando los alumnos y profesores.

Tenemos por un lado el registro de logs, los tags en forma de nube, de las páginas más visitadas, y por debajo observamos el número de actividad de cada equipo, y la tendencia que hay a navegar desde los equipos del aula, o desde MV o portátiles.

Éstos últimos, las MVs y los portátiles, son los únicos equipos de los que la única información que podemos sacar, es la que dejen navegando, por el proxy, ya que no tendremos agentes Beats instalados en ellos.



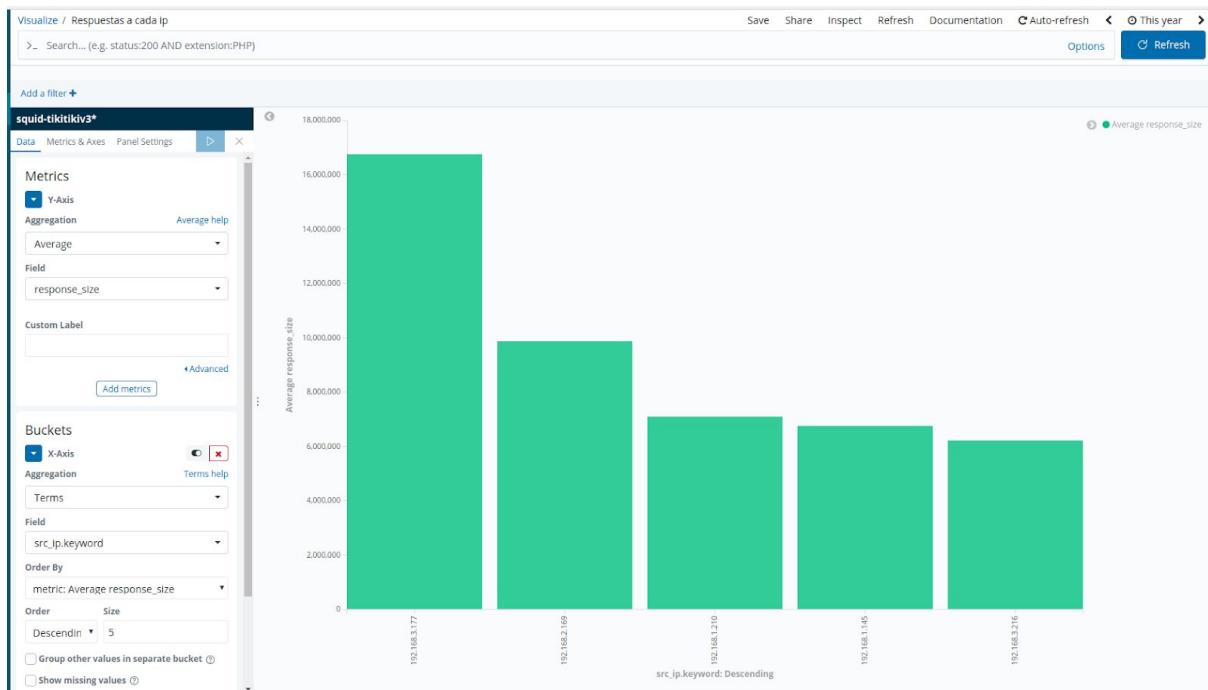
En el orden en el que observamos, primero vemos la tabla, que está basada en la siguiente búsqueda guardada:



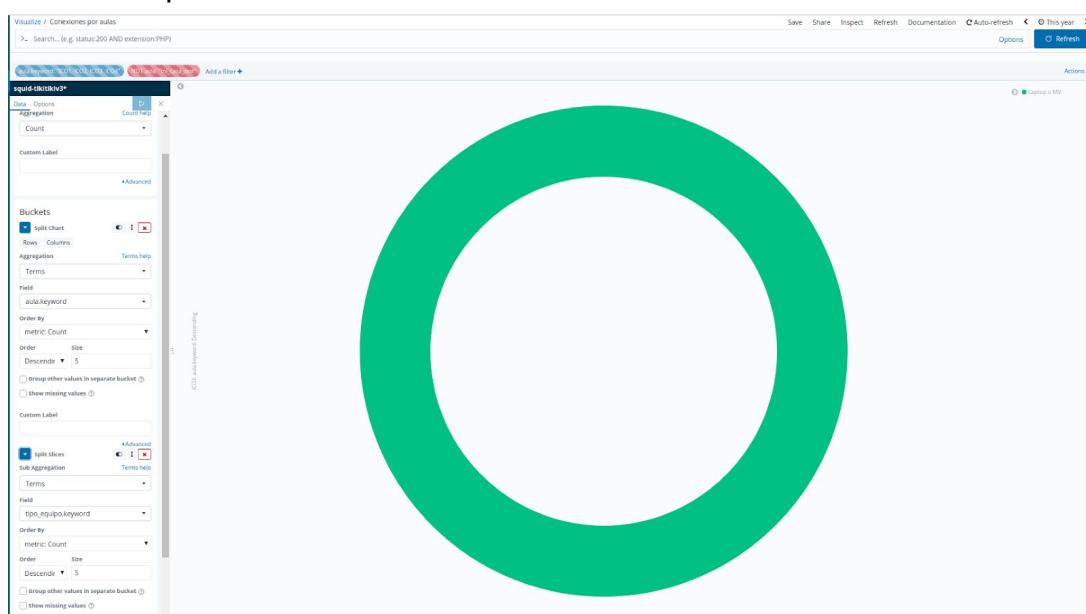
A la derecha observamos la nube de tags, que está construída de la siguiente manera:



Y ya en la última fila vemos el nivel de actividad de cada ip, realizado de la siguiente manera:

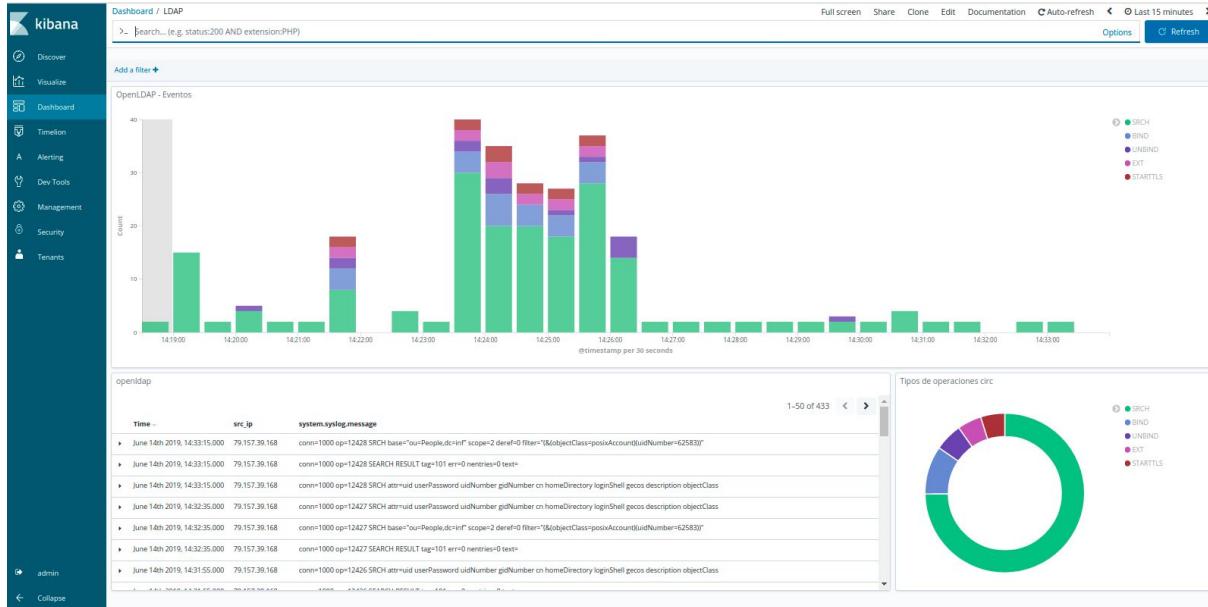


En cuanto a la tendencia a utilizar Laptop o MV,, en este caso, como esta diferenciación se hace en base a las IPs del departamento, y éstas pruebas se están realizando en remoto en el “entorno de test” las IPs que llegan al Elasticsearch no son las IPs privadas. Está llegando, para todos los equipos del departamento, la misma IP pública, por lo tanto en esta gráfica solo vemos “Laptop o MV”, porque todas las IPs detectadas, no corresponden con ninguna de las IPs privadas del departamento.

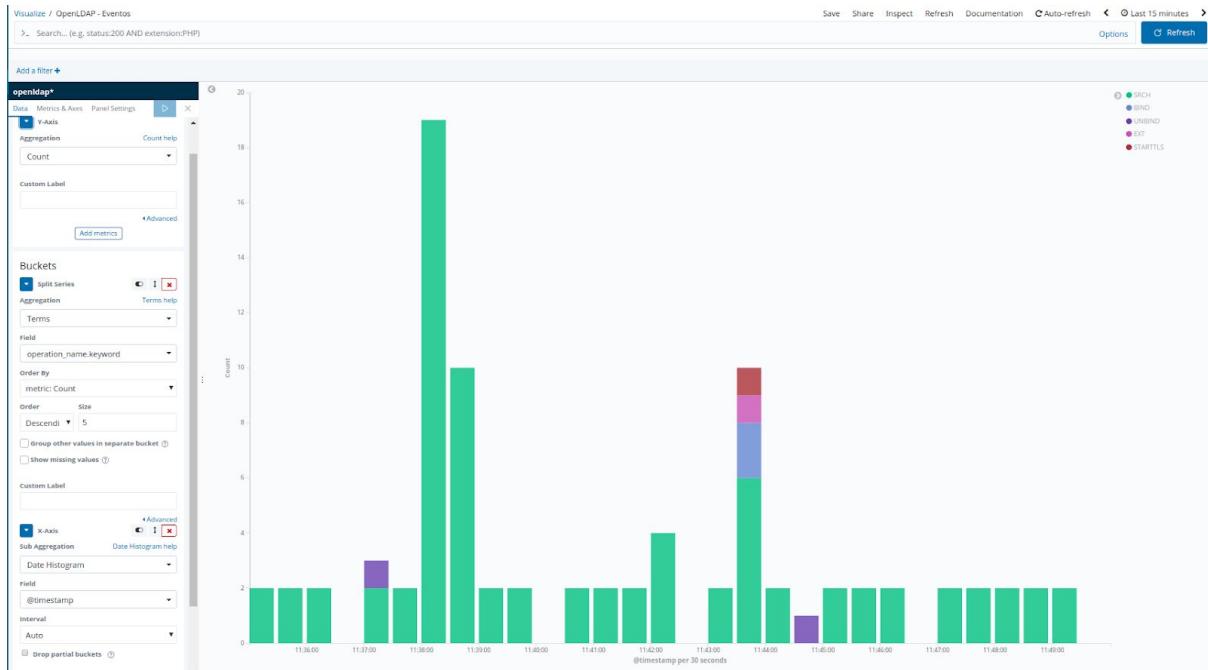


Panel LDAP

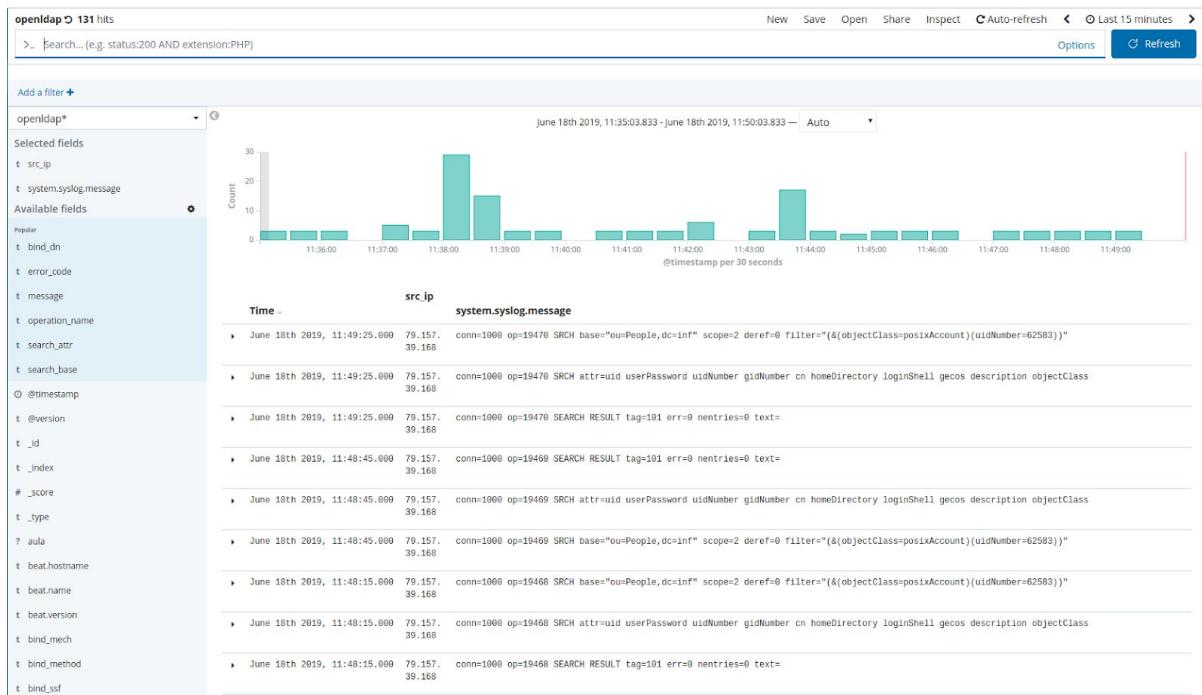
Este panel, independiente del anterior, nos muestra toda la información relevante a LDAP.



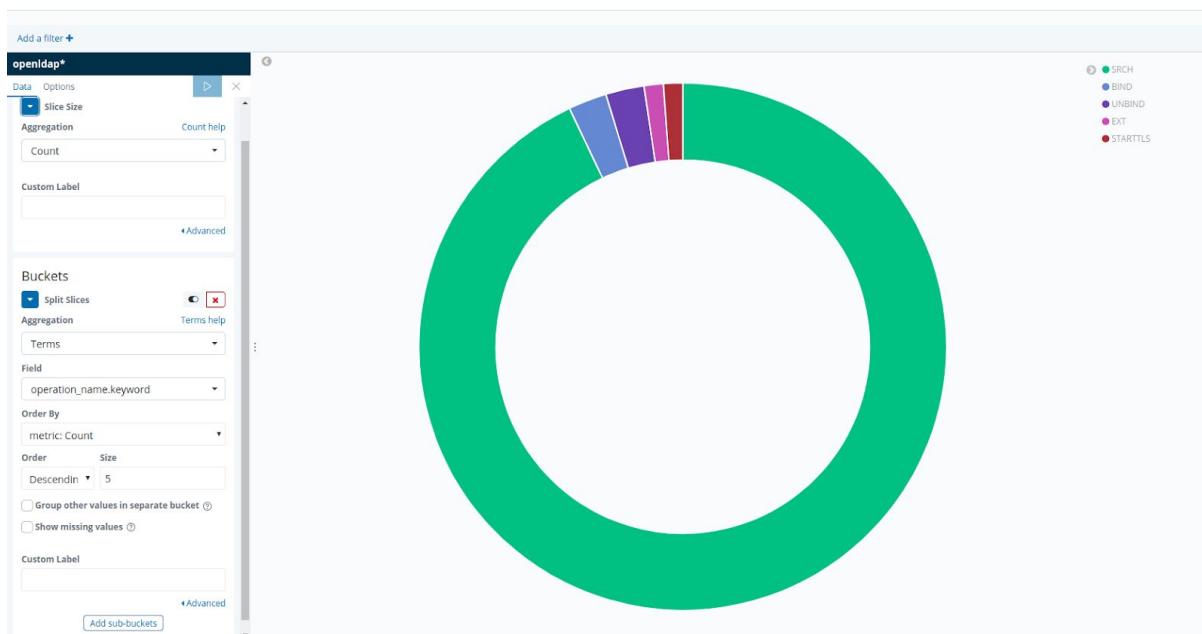
La gráfica superior está montada de la siguiente manera:



La vista de logs está basada en la siguiente búsqueda guardada:



Y la gráfica en forma de rueda está construída de la siguiente manera:

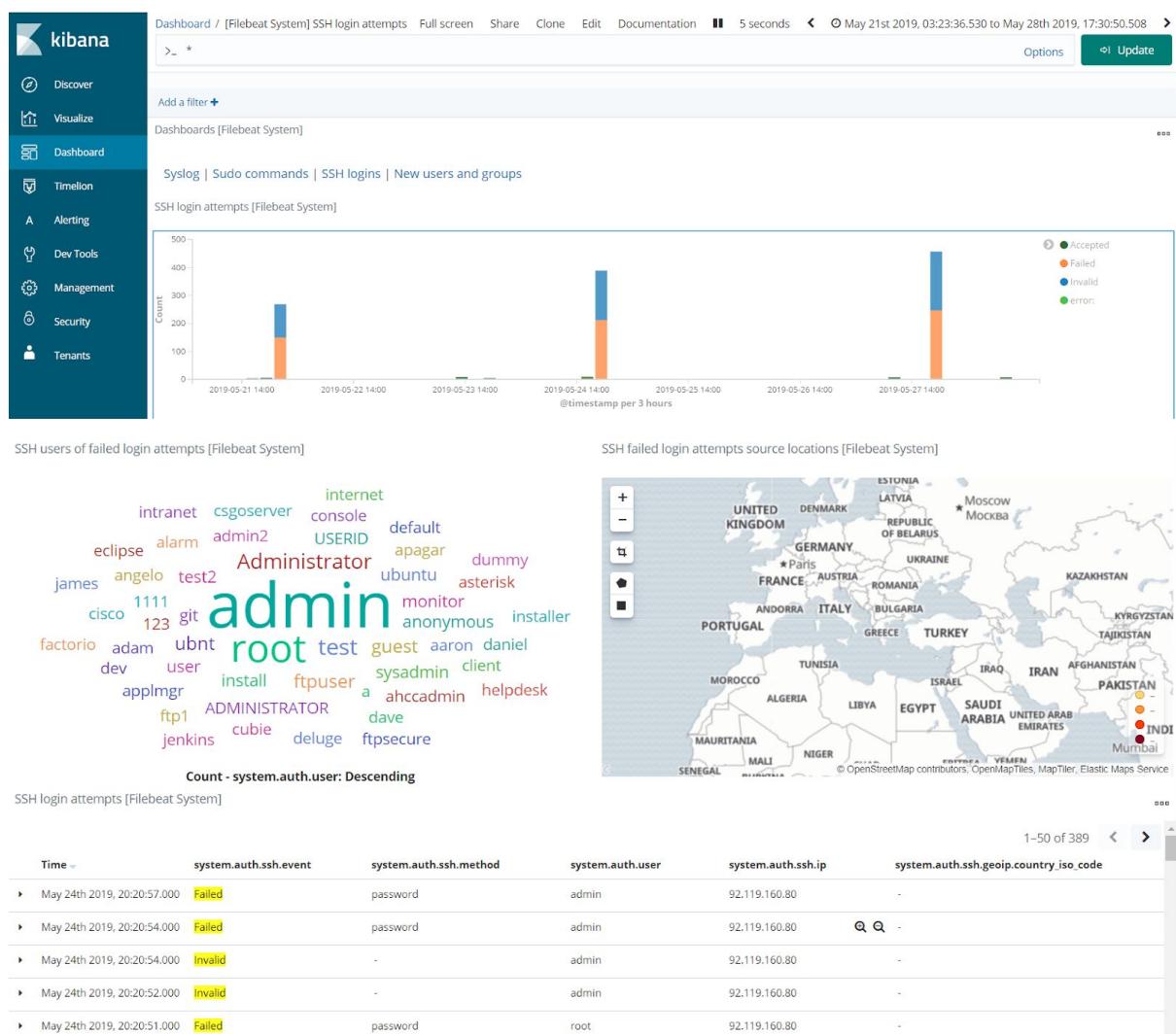


Curiosidades

Ataque detectado

En medio de la elaboración de éste proyecto, observamos cierta actividad inusual. Los días 21, 24, y 27 tuvieron lugar entre 300 y 400 intentos de inicio de sesión SSH contra una MV que utilizaba para pruebas personales de desarrollo. Todos los intentos inválidos o fallidos, y todos provenientes de una IP de origen Ruso.

Podemos observar en la nube de tags, los usuarios con los que intentaron conectarse. Claramente un ataque aleatorio de diccionario.



SSH users of failed login attempts [Filebeat System]



Si filtramos por accesos autorizados vemos que corresponden con Cádiz y Zaragoza. Las conexiones de Zaragoza deben de estar debidas al uso de la VPN de la empresa donde realizaba las FCT y adelantaba el proyecto.



Comparación Docker V/s Máquinas Virtuales

Se procedió a instalar XAMP en un Ubuntu Server 16.0 virtualizado con QUEMU/KVM para comparar el consumo de recursos de un contenedor Docker de XAMP.

Para realizar una prueba de estrés se ha utilizado un script de Python: “mobbage³⁸” Su implantación es muy sencilla:

```
sudo pip install mobbage
```

Generamos un fichero con el siguiente formato JSON apuntando al XAMPP del contenedor Docker y al XAMPP de la MV:

```
[  
  { "url": "http://localhost:8086", "count": 3000,  
   "url": "http://192.168.1.43", "count": 3000  
  }  
]
```

Y lanzamos el script:

```
mobbage -f test
```

Veremos por la salida de la consola:

```
evaristo@HermesTrismegisto:~$ mobbage -f test  
Starting mobbage with 1 worker.  
Results:  
  Total time:      00:00:10  
  Requests:        3000  
  Successes:       3000  
  Errors:          0  
  Availability:   100.00%  
  Minimum time:    1ms  
  Average time:    2ms  
  Maximum time:    15ms  
  Minimum size:    7.6KB  
  Average size:    7.6KB  
  Maximum size:    7.6KB  
  Total data:      22.7MB  
  Average data rate: 2.1MB/s  
  Concurrency:     274.49  
  
Results by return code  
  200:            3000
```

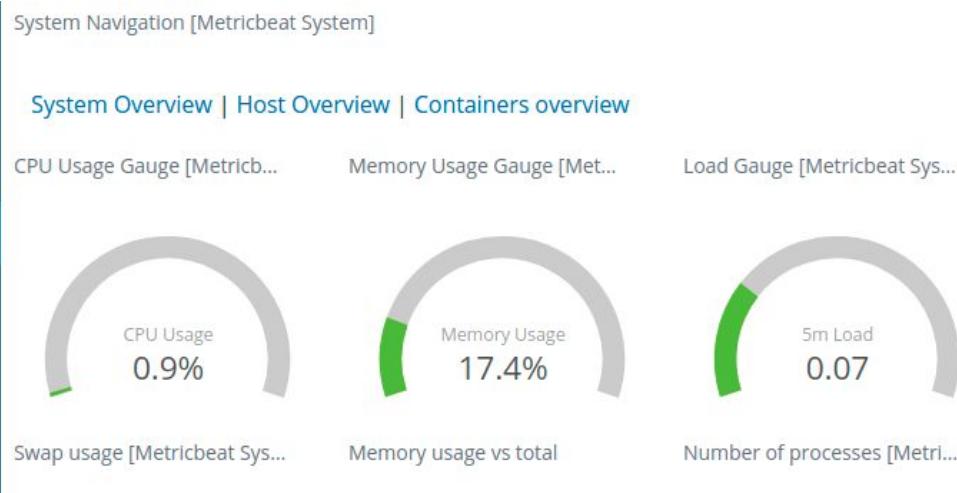
³⁸ Mobbage: <https://github.com/redfin/mobbage>

Y deberíamos de ver la actividad reflejada en las métricas:

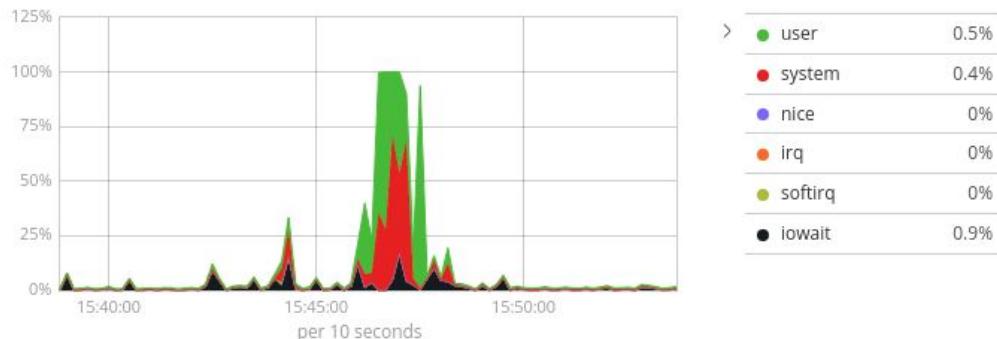


El resultado es el siguiente:

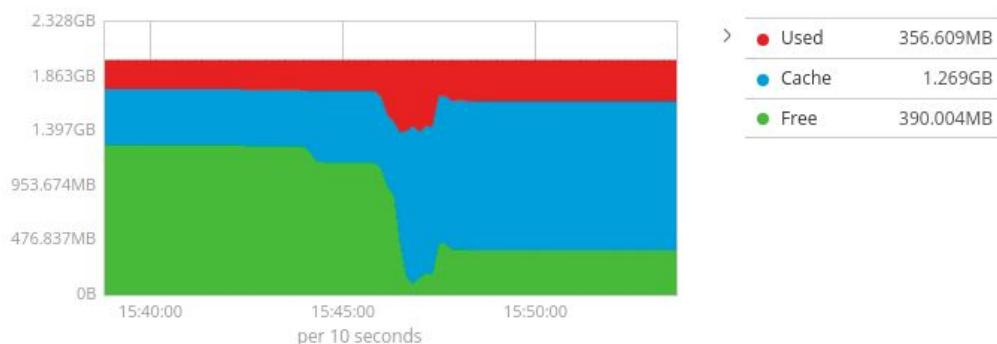
QUEMU/KVM:



CPU Usage [Metricbeat System]



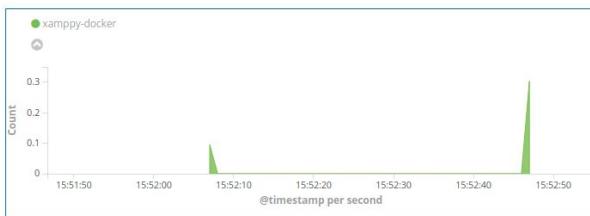
Memory Usage [Metricbeat System]



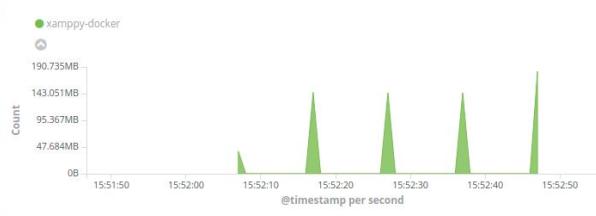
Contenedor Docker

Name	CPU usage (%)	DiskIO	Mem (%)	Mem RSS	Number of Containers
xamppy-docker	0.304	91.464	0.006	93.441MB	1

CPU usage [Metricbeat Docker]



Memory usage [Metricbeat Docker]



Conclusión de la comparación

La MV estaba consumiendo en ese momento 0,9 % de CPU y 17,4% de memoria mientras que el contenedor Docker consumía 0,30% de CPU y 0,006% de memoria

Un resultado anecdótico, para reafirmar el mínimo consumo de recursos de un contenedor Docker frente a la utilización de máquinas virtuales.

5. Conclusiones finales

El proyecto cuya documentación ya está llegando a su fin, es un proyecto que comenzó en con la búsqueda de un sistema para monitorear los hosts del departamento de informática del IES Fernando Aguilar Quignon, y que terminó siendo ésto y además un sistemas de análisis de los datos que diariamente circulan por las redes de dicho departamento, por lo que además de haber adquirido una gran cantidad de conocimiento sobre tecnologías que están a la orden del día en el campo de la monitorización y del análisis de datos, se ha cumplido con todos los objetivos marcados en un principio.

Como propuestas de ampliaciones futuras del sistema implantado se sugiere:

- Beat “Journalbeat”³⁹
 - Journalbeat, recoge logs de journald, se han realizado varias pruebas con él pero a día de hoy todavía está en una momento inmaduro de su desarrollo y no es estable, pero seguramente en un futuro será completamente funcional, ya que tiene actividad semanal en Github.
- Dockerización⁴⁰ del entorno completo y así como de sus agentes.
 - Si bien es cierto que la documentación está disponible a día de hoy, aún hay muchos problemas de conectividad, que se reflejan en el foro de la comunidad de Opendistro, pero con un poco más de tiempo seguramente sea 100% funcional.
- Monitorización de portátiles mediante los logs de DHCP.
 - Mediante estos registros se podría llegar a conseguir rastrear los equipos portátiles de los alumnos que están en el aula.

³⁹ Journalbeat: <https://github.com/elastic/beats/tree/master/journalbeat>

⁴⁰ Doc docker: <https://opendistro.github.io/for-elasticsearch-docs/docs/install/docker/>

Anexos

Anexo 1

Alternativas de código abierto y gratis a X-Pack por componente:

Elasticsearch Security

- SearchGuard:
 - Proporciona una alternativa gratuita y de código abierto para X-Pack Security. El soporte de SearchGuard y las funciones de la empresa no son gratuitos (el modelo de licencia es por grupo), pero probablemente sea un ahorro de costos en relación con X-Pack.
 - <https://github.com/floragunncom/search-guard>
- ReadonlyREST:
 - ReadonlyREST es un complemento ligero de Elasticsearch que agrega encriptación, autenticación, autorización y capacidades de control de acceso a la API REST incorporada de Elasticsearch. El núcleo de este complemento es un motor de ACL que verifica cada solicitud entrante a través de una secuencia de reglas un poco como un firewall. Hay una docena de reglas que se pueden agrupar en secuencias de bloques y formar una representación poderosa de una cadena lógica.
 - El complemento Elasticsearch conocido como ReadonlyREST Freese está publicado bajo la licencia GPLv3, o alternativamente, y también tiene una licencia comercial. A diferencia del complemento Elasticsearch, los complementos de Kibana son solo comerciales.
 - <https://github.com/beshu-tech/readonlyrest-docs/blob/master/elasticsearch.md>

Elasticsearch Alerting

- Elastalert:
 - Es una herramienta simple y popular de código abierto para alertar sobre anomalías, picos u otros patrones de interés encontrados en los datos almacenados en Elasticsearch. Elastalert funciona con todas las versiones de Elasticsearch.
 - <https://github.com/yelp/elastalert>
- Logagent:
 - Es un cargador de registro general de código abierto. Sin embargo, puede programar consultas de Elasticsearch (entrada), filtrar los resultados utilizando criterios personalizados y alertar a través de

salidas conectables como Slack. Por lo tanto, usar Logagent para alertar sobre los datos de Elasticsearch es solo una cuestión de configuración .

- <https://github.com/sematext/logagent-js>
- Sentinel:
 - extiende Kibana con la funcionalidad de monitorear, notificar e informar sobre los cambios en las series de datos mediante consultas estándar, validadores programables y una variedad de acciones configurables.
 - <https://github.com/sirensolutions/sentinel>

Elasticsearch Reporting

- Skedler:
 - Proporciona una programación sencilla de los informes en PDF, XLS y PNG para los paneles de Kibana. Los planes de pago son sólo unos pocos cientos de dólares por año.
 - <https://www.skedler.com/>
- Sentinel:
 - También tiene capacidades de "Informes" programadas (instantáneas PNG / PDF).
 - <https://github.com/sirensolutions/sentinel>

Alternativas de Elasticsearch Graph

- Kibi es una bifurcación que extiende Kibana con un modelo de datos relacionales y la capacidad de hacer uniones sobre múltiples índices. Además, soporta datos relacionales de bases de datos SQL. La edición para empresas incluye visualización de gráficos, alertas e informes, funciones de seguridad, componentes adicionales y soporte.
- <https://github.com/sirensolutions/kibi>
- Kbn_network:
 - Complemento gratuito de código abierto para Kibana 5 pensado para visualización en red.
 - https://github.com/dlumbre/kbn_network
- DIY: Cytoscape.js , Visjs.org (código abierto)
 - Se trata principalmente de la programación frontend de JavaScript y de la conversión de los resultados de las consultas de Elasticsearch a una estructura de gráfico (nodos y bordes). Hay varias bibliotecas de visualización de gráficos de código abierto para representar estructuras de datos de gráficos en el navegador.
 - <http://js.cytoscape.org/>
 - <https://visjs.org/>

Elasticsearch Machine Learning

- Knowi es una herramienta de inteligencia empresarial que soporta de forma nativa muchas fuentes de datos SQL y NoSQL, incluida Elasticsearch. Knowi recientemente agregó capacidades de aprendizaje automático , combinando BI y AI en una sola plataforma, para respaldar el análisis predictivo y prescriptivo.
- <https://www.knowi.com/>

Anexo 2

Historia completa de ELK Stack

Principios de los años 2000

Comenzó con una aplicación de recetas de cocina.

En un departamento de Londres, Shay Banon estaba buscando trabajo mientras su esposa asistía a la escuela de cocina en Le Cordon Bleu. En su tiempo libre, comenzó a construir un motor de búsqueda para su creciente lista de recetas.

Su primera iteración fue llamada Compass. El segundo fue Elasticsearch (con Apache Lucene debajo del capó). Elasticsearch, de código abierto, creó el canal IRC #elasticsearch y esperó a que los usuarios aparecieran.

La respuesta fue impresionante. Los usuarios lo tomaron de forma natural y fácil. La adopción se disparó, se formó una comunidad y la gente se dio cuenta: Steven Schuurman, Uri Boness y Simon Willnauer. Juntos, fundaron una empresa de búsqueda.



Junio 2012

En la época en que se fundó Elasticsearch Inc., otros dos proyectos de código abierto estaban tomando vuelo.

Jordan Sissel estaba trabajando en Logstash, una herramienta de ingestión de código abierto y conectable para enviar registros al "aliño" que el usuario elegiría, uno de los cuales fue Elasticsearch. También estaba desarrollando una interfaz de usuario para visualizar datos de registro, y en el mejor de los casos era inestable.

Afortunadamente, alguien más estaba jugando con el desafío de visualización. Rashid Khan, que estaba trabajando en una interfaz de usuario de código abierto llamada Kibana.

Shay, Jordan y Rashid se conocían entre ellos y sus proyectos desde hacía un tiempo y decidieron formar un equipo, lo que dio como resultado el ELK Stack: Elasticsearch, Logstash y Kibana Stack.

Un poco después, lanzaron dos complementos comerciales: Marvel para monitoreo y Shield para seguridad.

Julio 2015

En los primeros días, la creación y el lanzamiento de software en Elastic se adoptaron enfoques de "todos los ingenieros": "*envíe la versión que desee, siempre que lo deseé, solo hágala increíble.*"

Kibana tenía betas, Logstash tenía hitos, Elasticsearch tenía números. Los plugins pasaron a su antojo. Fue caótico, pero funcionó ... hasta que no lo hizo.

Como los usuarios estaban haciendo más con el producto, necesitaban crear un producto que hiciera más por los usuarios. Se añadieron más funciones, se enviaron más solicitudes de extracción, se crearon nuevos complementos y extensiones. La maravilla aumentó, la complejidad emergió y las cosas se complicaron.

Por ejemplo, si se ejecutaba la versión 1.7 de Elasticsearch y la versión 2.3 de algún plugin, no había una forma automática de saber si eran compatibles o si el plugin estaba fallando en silencio. Esto era un error.

Haciendo una pausa para tomar un Beat (s)

Mientras los equipos de productos luchaban con los números de versión, se desarrollaba otra historia del producto. En 2015, se le dió la bienvenida a Packetbeat, un equipo de marido y mujer con sede en Berlín que diseñó una manera ligera de enviar datos de red a Elasticsearch, a la familia Elastic.

Eso les hizo pensar: *¿qué pasaría si tuviéramos una familia de remitentes de datos ligeros y de un solo propósito para enviar datos de red, registros, métricas, datos de auditoría y mucho más de las máquinas de borde a Logstash y Elasticsearch?* Y así nació Beats.

Octubre 2015

Octubre de 2015 marcó un punto de inflexión para abordar las complejidades de compatibilidad y versión de nuestros productos.

Apodado el "lanzamiento de bonanza", fue la primera vez que todos los productos (Elasticsearch 2.0, Logstash 2.0, Watcher 2.0, Shield 2.0 y Kibana 4.2) se enviaron juntos el mismo día, con la misma versión (Beats 1.0 tenía otro mes para hornear.)

Octubre 2016

Alinear la cadencia de lanzamiento con Elasticsearch 2.0 fue el primer paso hacia una oferta de productos más madura. El lanzamiento 5.0 fue el segundo paso. Introdujo una experiencia de inicio más integrada, mejor probada y más fácil que nunca.

La versión 5.0 también incluía todos complementos comerciales (que en su momento llamábamos Shield, Marvel y Watcher) en una sola extensión llamada X-Pack. Consistía en características como seguridad, monitoreo y alertas para nuestros productos principales, y creció para incluir el aprendizaje automático cuando se incorporó a la empresa Elastic una empresa con sede en Londres llamada Prelert.

En la versión 5.3 (lanzada en marzo de 2017), Filebeat introdujo formalmente el concepto de "módulos" o un conjunto de configuraciones seguras para enviar, analizar, almacenar, analizar y visualizar formatos de registro comunes (por ejemplo, Apache, Nginx, MySQL, etc.) .) En la pila elástica. Los módulos simplificaron la experiencia inicial de ir del conjunto de datos al panel de control.

Metricbeat y Packetbeat tenían sus propios sabores de módulos, y meses después, Logstash presentaría módulos propios para los datos de ArcSight y NetFlow.

Mayo 2017

Nace Elastic Cloud Enterprise (o ECE) para permitir que las empresas grandes y pequeñas confíen en el servicio alojado de Elastic.

Octubre 2017

Soluciones elásticas precipitadas

A medida que los módulos comenzaron a multiplicarse, comenzar a utilizar Elastic Stack para abordar un caso de uso particular, como el registro o las métricas, se hizo cada vez más fácil. Y el impulso continuó creciendo cuando se unieron con Opbeat, una empresa de monitoreo de rendimiento de aplicaciones (APM) con sede en Copenhague, y Swiftype, una empresa de búsqueda de empresas y sitios con sede en San Francisco, unos meses más tarde. Ambas empresas pasaron a formar parte de Elastic.

Junio 2018

Abriendo el código X-Pack

Se abre el código de X-Pack

Todas las características de X-Pack ahora se distribuyen con las distribuciones predeterminadas de Elasticsearch, Kibana, Beats y Logstash. Este cambio no quitó ningún código de Apache 2.0.

Octubre 2018

Exactamente a las 9:30 am, hora del este el 5 de octubre, sonó el timbre de la Bolsa de Nueva York, que oficialmente convirtió a Elastic en una empresa pública. Con un récord de 230 Elasticians en el piso de operaciones y cientos más en todo el mundo, la compañía distribuida celebró alcanzar este notable hito.

6. Bibliografía

Acerca de la monitorización

La monitorización

<https://www.servotic.com/monitorizacion-de-servidores-por-que-necesitamos-un-sistema-de-monitorizacion/>

https://es.wikipedia.org/wiki/Monitoreo_de_red

<https://es.wikipedia.org/wiki/Monitorizaci%C3%B3n>

Sitios web de herramientas

Sítio web de Nagios:

<https://www.nagios.com/>

Sítio web de pandora:

<https://pandorafms.com/es/>

Zabbix:

<https://www.zabbix.com/>

Sítio web de Check_MK

<https://checkmk.de/>

Sitio web de Splunk: <https://www.splunk.com>

Acerca de Check_MK

Plugin para habilitar Telegram como método de alerta en Check_MK implementado durante la FCT:
<https://suevaristo.blogspot.com/2019/04/bot-telegram-para-recibir.html>

Script/agente customizado para monitorizar el estado de las alertas de Amazon CloudWatch implementado durante la FCT:
<https://suevaristo.blogspot.com/2019/05/monitoreando-la-monitorizacion.html>

Check_MK

<https://mathias-kettner.com/cms.html>

<https://check-mk-documentation.readthedocs.io/en/latest/intro.html>

<https://paulgraydon.co.uk/posts/2012-09-20-moving-from-zabbix-to-check-mk/>

http://mathias-kettner.com/checkmk_ways_to_install.html

http://mathias-kettner.com/checkmk_livestatus.html

http://mathias-kettner.com/checkmk_multisite.html

http://mathias-kettner.com/checkmk_wato.html

http://mathias-kettner.com/checkmk_flexible_notifications.html

http://mathias-kettner.com/checkmk_bi.html

<https://check-mk-documentation.readthedocs.io/en/latest/cmkarchitecture.html>

https://mathias-kettner.de/cms_wato_monitoringagents.html

https://mathias-kettner.com/cms_graphing.html.

https://mathias-kettner.de/cms_notifications.html

https://mathias-kettner.com/cms_monitoring_basics.html

Check_MK a Grafana

<https://truepath.zendesk.com/hc/en-us/articles/115004758503-Exporting-Check-MK-Performance-Data-to-Grafana>

Formación Check_MK:

https://mathias-kettner.de/cms_training_ckm1.html

https://mathias-kettner.de/cms_training_ckm2.html

mk_logwatch

https://mathias-kettner.de/checkmk_logfiles.html

<http://zoomadmin.com/HowToInstall/UbuntuPackage/check-mk-agent-logwatch>

https://mathias-kettner.com/checkmk_logfiles.html

Acerca de Amazon Elasticsearch

Utilizar Amazon S3 para almacenar un único índice de servicios de Amazon Elasticsearch:

<https://aws.amazon.com/es/blogs/database/use-amazon-s3-to-store-a-single-amazon-elasticsearch-service-index/>

Elastic - Requisitos de hardware:

<https://www.elastic.co/guide/en/elasticsearch/guide/current/hardware.html>

Acerca de ELK Stack

ELK Stack

<https://www.elastic.co/elk-stack>

<https://github.com/deviantony/docker-elk>

<https://logz.io/learn/complete-guide-elk-stack/#intro>

<https://dzone.com/articles/running-data-analytics-on-application-events-and-l>

<https://logz.io/blog/server-log-analysis/>

https://github.com/elastic/examples/tree/master/Common%20Data%20Formats/apache_logs

<https://howtodoinjava.com/microservices/elk-stack-tutorial-example/>

<https://medium.com/@brunoamaroalmeida/enabling-centralized-application-logging-using-the-elastic-elk-stack-from-stratch-a-15-minutes-eba501230b3d>

Alternativas a X-Pack

<https://sematext.com/blog/x-pack-alternatives/>

<https://www.linkedin.com/pulse/cheaper-alternatives-elastic-x-pack-work-just-well-do-uglas-miller/>

<https://github.com/ElasticHQ/elasticsearch-HQ>

Search guard

<https://github.com/floragunncom/search-guard>

ReadonlyREST

<https://readonlyrest.com/download/>

Análisis de red Wireshark + ELK Stack

<https://www.elastic.co/es/blog/analyzing-network-packets-with-wireshark-elasticsearch-and-kibana>

Fix logstash en Open Distro

<https://discuss.opendistrocommunity.dev/t/logstash-setting/197>

Beats:

<https://github.com/elastic/beats>

Buscar software ELK - OSS:

<https://www.elastic.co/es/downloads/past-releases>

journalist

<https://www.elastic.co/guide/en/beats/journalbeat/current/journalbeat-getting-started.html>

Explicación código abierto X-Pack:

<https://www.elastic.co/es/products/x-pack/open>

los prospectores son ahora los input

<https://www.elastic.co/es/blog/brewing-in-beats-rename-filebeat-prospectors-to-inputs>

Jugando con el source

<https://medium.com/tensult/log-centralization-using-filebeat-and-logstash-11640f77cf70>

Segurizar filebeat y logstash

<https://www.elastic.co/guide/en/beats/filebeat/current/configuring-ssl-logstash.html>

cómo funciona filebeat

<https://www.elastic.co/guide/en/beats/filebeat/current/how-filebeat-works.html>

Patrones GROK

<https://github.com/elastic/logstash/blob/v1.4.2/patterns/grok-patterns>

GROK CONSTRUCTOR

Solo para referencias:

<https://grokconstructor.appspot.com/>

Muy bueno para debug:

<http://grokdebug.herokuapp.com/>

Filebeat y Logstash

<https://logz.io/blog/filebeat-vs-logstash/>

Ingesta de varios Filebeat a un logstash

<https://discuss.elastic.co/t/how-to-tag-log-files-in-filebeat-for-logstash-ingestion/4471/3>

config de Filebeat

<https://www.elastic.co/guide/en/beats/filebeat/1.1/configuration-filebeat-options.html#configuration-fields>

Deploying-and-scaling Logstash

<https://www.elastic.co/guide/en/logstash/current/deploying-and-scaling.html>

ELK vs graylog

<https://coralogix.com/log-analytics-blog/log-management-comparison-elk-vs-graylog/>

Auth GROK

<http://www.alexlinux.com/logstash-ssh-login-example/>

libbeat - Framework f

<https://github.com/elastic/beats/tree/master/libbeat>

Traducir nombres:

<https://www.elastic.co/guide/en/logstash/current/pipeline.html>

Los filtros:

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

Acerca de la Arquitectura Elasticsearch:

<https://buildingvts.com/elasticsearch-architectural-overview-a35d3910e515>
<http://solutionhacker.com/elasticsearch-architecture-overview/>
<https://blog.insightdatascience.com/anatomy-of-an-elasticsearch-cluster-part-i-7ac9a13b05db>
<https://www.elastic.co/es/blog/how-many-shards-should-i-have-in-my-elasticsearch-cluster>
<https://www.elastic.co/guide/en/elasticsearch/guide/current/inside-a-shard.html>
<https://www.elastic.co/guide/en/elasticsearch/guide/current/document.html>
https://www.elastic.co/guide/en/elasticsearch/guide/current/_document_metadata.html

Diseñando el clúster perfecto de Elasticsearch

<https://thoughts.t37.net/designing-the-perfect-elasticsearch-cluster-the-almost-definitive-guide-e614eabc1a87>

Acerca de Logstash:

<https://www.elastic.co/guide/en/logstash/6.7/logstash-config-for-filebeat-modules.html#logstash-config-for-filebeat-modules>
<https://www.elastic.co/es/blog/a-practical-introduction-to-logstash>

Acerca de OpenDistro for Elasticsearch:

<https://opendistro.github.io/for-elasticsearch/>
<https://aws.amazon.com/es/blogsopensource/running-open-distro-for-elasticsearch/>
<https://aws.amazon.com/es/blogs/aws/new-open-distro-for-elasticsearch/>
<https://aws.amazon.com/es/blogsopensource/keeping-open-source-open-open-distro-for-elasticsearch/>
https://medium.com/@maxy_ermayank/tl-dr-aws-open-distro-elasticsearch-fc642f0e592a
<https://opendistro.github.io/for-elasticsearch/downloads.html>
<http://diego-pacheco.blogspot.com/2019/03/running-aws-es-open-distro-locally.html>
<https://medium.com/@navatm/tutorial-elasticsearch-kibana-secured-and-cheap-76fd63a7597>
<https://discuss.opendistrocommunity.dev/t/why-no-attribution-to-searchguard/131>

Agregue sus propios certificados SSL para abrir Distro para Elasticsearch:

<https://aws.amazon.com/es/blogsopensource/add-ssl-certificates-open-distro-for-elasticsearch/>

Protocolo del leñador

<https://github.com/elastic/logstash-forwarder/blob/master/PROTOCOL.md>

<https://www.npmjs.com/package/lumberjack-protocol>

Características de la arquitectura de Elasticsearch

<https://openwebinars.net/blog/caracteristicas-de-la-arquitectura-de-elasticsearch/>

Squid y Logstash:

<https://medium.com/@thomasdecaux/analyze-web-traffic-with-squid-proxy-elasticsearch-logstash-kibana-stack-e2a471e34bc4>

Creación de módulos de filebeat detallada en:

<https://www.elastic.co/guide/en/beats/devguide/current/filebeat-modules-devguide.html>

<https://www.elastic.co/es/blog/monitoring-applications-with-elasticsearch-and-elastic-apm>

AWS Elasticsearch:

https://docs.aws.amazon.com/es_es/elasticsearch-service/latest/developerguide-sizing-domains.html

Guerra empresarial AWS Vs Elastic:

<https://code972.com/blog/2019/03/116-dont-confuse-awss-open-distro-for-elasticsearch-with-altruism>

Otros

Boots Telegram

<https://orekait.com/blog/bot-en-telegram/>

FAQ

de

Prometheus:

<https://prometheus.io/docs/introduction/faq/#how-to-feed-logs-into-prometheus>

Limitaciones de V. Community

<https://pandorafms.com/es/precios-de-pandora-fms/#pricing%7C1>

Machine learning para nginx

<https://www.elastic.co/es/blog/machine-learning-for-nginx-logs>

Comparación Splunk Vs ELK Stack:

<https://devops.com/splunk-elk-stack-side-side-comparison/>

Grafana Vs Kibana:

<https://logz.io/blog/grafana-vs-kibana/>

Prometheus y Soundcloud:

<https://developers.soundcloud.com/blog/prometheus-monitoring-at-soundcloud>

Contenedores splunk y squid

<https://github.com/brunoamaroalmeida/awesome-quickstart-containers>

Gráfico de puntos fuertes:

Creado con <https://livegap.com/charts/>

tf – idf

<https://en.wikipedia.org/wiki/Tf%E2%80%93idf>

Recolector de basura:

https://es.wikipedia.org/wiki/Recolector_de_basura

Yourkit - Java

<https://www.yourkit.com/java/profiler/features/>

MurmurHash

<https://en.wikipedia.org/wiki/MurmurHash>

CRUD:

<https://es.wikipedia.org/wiki/CRUD>

Ingest node:

<https://www.elastic.co/guide/en/elasticsearch/reference/master/ingest.html>

KVM dominio de la memoria y el uso del disco:

<https://www.redhat.com/archives/libvir-list/2014-February/msg01703.html>