

# IES Fernando Aguilar Quignon

CFGS ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED



PROYECTO DE ADMINISTRACIÓN DE SISTEMAS  
INFORMÁTICOS EN RED

CENTRALIZACIÓN Y ANÁLISIS DE LOGS MEDIANTE  
OPENSEARCH

ALUMNO/A: Pedro Peralta Guerrero  
TUTOR/A: Javier García Estévez

CONVOCATORIA JUNIO | CURSO ACADÉMICO 2022/23



## Resumen

El problema que se busca resolver con este proyecto es el de la falta de una plataforma integrada y eficiente para el almacenamiento, procesamiento y visualización de los logs generados por los diferentes sistemas y por el servicio OpenLDAP. Actualmente, cada sistema tiene su propio método de registro y almacenamiento de logs, lo que dificulta el seguimiento, la depuración y el análisis de los datos.

El objetivo de este proyecto es diseñar e implementar una solución de centralización y análisis de logs que permita recopilar, almacenar, procesar y visualizar los logs de todos los sistemas y del servicio OpenLDAP del instituto, concretamente de los talleres de ICO, en una sola plataforma.

Para lograr esta solución se hará uso de OpenSearch(derivado de Elasticsearch) ya que nos permite a las personas ingerir, asegurar, buscar, agregar, ver y analizar datos fácilmente. También se hará uso de OpenSearch Dashboards, derivado de Kibana, se trata de la interfaz de usuario que le permite visualizar datos de OpenSearch. Además de hacer uso de OpenSearch y OpenSearch Dashboards, será necesario Logstash ya que gracias a este podremos procesar los eventos en tiempo real. Por otro lado, será imprescindible el uso de algún cliente para enviar los logs y para ello se usará Filebeat un cliente ligero para reenviar y centralizar logs y archivos.

El proyecto ha sido realizado en colaboración con el tutor para poder entender sus necesidades ya que eran indispensables para la correcta elaboración de la solución final.

Como futuras líneas de investigación, se sugieren los siguientes temas relacionados con las herramientas utilizadas en este proyecto de centralización y análisis de logs:

- Explorar en mayor profundidad las capacidades de optimización de consultas en OpenSearch. Investigar técnicas y estrategias avanzadas para mejorar el rendimiento y la eficiencia de las consultas, como el uso de índices personalizados, ajuste de parámetros y optimización de consultas complejas.
- Se podría investigar la integración de OpenSearch con otros sistemas y servicios relacionados. Por ejemplo, explorar cómo aprovechar los datos de logs almacenados en OpenSearch para mejorar la detección de amenazas en sistemas de seguridad perimetral o integrar con herramientas de análisis de big data para obtener una visión más completa de los patrones de comportamiento.
- Otra área de investigación relevante podría ser el desarrollo de técnicas de monitoreo y alertas en tiempo real más avanzadas. Investigar cómo configurar alertas basadas en eventos específicos en los logs y explorar opciones para la detección automática de anomalías o patrones inusuales de comportamiento, lo que permitiría una respuesta más rápida a incidentes y problemas.
- Por último, se sugiere investigar en el campo del análisis avanzado de logs utilizando técnicas como el aprendizaje automático y la inteligencia artificial. Explorar algoritmos y modelos para la detección de patrones, el análisis de comportamiento y la correlación de eventos, lo que podría revelar conocimientos más profundos y valiosos sobre los datos de logs almacenados en OpenSearch.

Resumiendo, gracias a este proyecto se posibilitará una nueva manera para poder centralizar y analizar logs de los sistemas y del servicio OpenLDAP integrado en las aulas de ICO del instituto IES Fernando Aguilar Quignon.



## Índice

<b>Resumen</b>	<b>1</b>
<b>1. Capítulo I: Introducción</b>	<b>4</b>
1.1. Contextualización o Situación Inicial . . . . .	4
1.2. Alcance del proyecto . . . . .	4
1.3. Objetivos del proyecto . . . . .	4
1.4. Planificación del proyecto . . . . .	5
1.4.1. Desglose de tareas. . . . .	5
1.4.2. Recursos necesarios . . . . .	5
1.4.3. Comunicación y seguimiento . . . . .	6
<b>2. Capítulo II: Marco Teórico</b>	<b>7</b>
2.1. Syslog & Auth.log . . . . .	7
2.1.1. Syslog . . . . .	7
2.1.2. Auth.log . . . . .	7
2.2. OpenLDAP . . . . .	8
2.2.1. Funcionamiento . . . . .	8
2.2.2. Tipos de operación . . . . .	9
2.3. Docker . . . . .	10
2.4. OpenSearch . . . . .	11
2.5. OpenSearch Dashboards . . . . .	12
2.6. Logstash . . . . .	13
2.7. Filebeat . . . . .	14
<b>3. Capítulo III: Análisis y Resultados</b>	<b>15</b>
3.1. Análisis de requisitos . . . . .	15
3.1.1. OpenSearch & OpenSearch Dashboards & Logstash . . . . .	15
3.1.2. Filebeat . . . . .	15
3.2. Diseño de la solución . . . . .	16
3.2.1. Arquitectura propuesta para la centralización y procesamiento de logs . . . . .	16
3.2.2. Configuraciones previas del sistema . . . . .	16
3.2.3. Configuración de OpenSearch . . . . .	16
3.2.4. Configuración de Logstash . . . . .	18
3.2.5. Configuración de Filebeat . . . . .	23
3.2.6. Configuración de OpenSearch Dashboards . . . . .	24
3.3. Implementación . . . . .	27
3.3.1. Descripción de las etapas y pasos seguidos durante la implementación de la solución. . . . .	27
3.3.2. Mención de posibles desafíos o problemas encontrados durante la implementación y cómo se resolvieron. . . . .	28
3.4. Resultados . . . . .	29
3.4.1. Antes y Después . . . . .	29
<b>4. Capítulo IV: Conclusiones y Líneas Futuras</b>	<b>31</b>
<b>Referencias.</b>	<b>32</b>
<b>Anexo A: Manual de instalación.</b>	<b>34</b>
<b>Anexo B: Manual para el administrador.</b>	<b>38</b>



## Índice de figuras

1.	Logotipo de OpenLDAP . . . . .	8
2.	Árbol LDAP. . . . .	9
3.	Logotipo de Docker . . . . .	10
4.	Logotipo de OpenSearch . . . . .	11
5.	Arquitectura propuesta . . . . .	16
6.	OpenSearch: Índices . . . . .	24
7.	OpenSearch: Index Pattern 2 . . . . .	24
8.	OpenSearch: Index Pattern 2 . . . . .	25
9.	OpenSearch: Discover . . . . .	25
10.	OpenSearch: Dashboard . . . . .	26
11.	OpenSearch: Visualizaciones . . . . .	26
12.	Antes: Consulta zgrep . . . . .	29
13.	OpenSearch: Presentación final de los logs . . . . .	30
14.	Anexo A: Error dashboard . . . . .	35
15.	Anexo A: Comprobación de contenedores . . . . .	36
16.	Anexo A: Fichero filebeat.yml . . . . .	37
17.	Anexo B: Manejo de índices . . . . .	38
18.	Anexo B: Patrones de índices . . . . .	39
19.	Anexo B: Discover . . . . .	40
20.	Anexo B: Dashboard . . . . .	41
21.	Anexo B: Crear una visualización . . . . .	42
22.	Anexo B: Ejemplo de gráfica . . . . .	42



## **1. Capítulo I: Introducción**

### **1.1. Contextualización o Situación Inicial**

El Instituto IES Fernando Aguilar, concretamente el departamento de informática, se enfrenta al desafío de gestionar los logs generados por los diferentes sistemas y el servicio OpenLDAP en las aulas de informática. Actualmente, cada sistema tiene su propio método de registro y almacenamiento de logs, lo que dificulta la capacidad de hacer un seguimiento, depurar y analizar eficientemente los datos de los logs.

El entorno de este proyecto se centra en la infraestructura tecnológica del instituto, que incluye los sistemas utilizados en las aulas de informática y el servidor LDAP para la autenticación. El objetivo es mejorar la eficiencia y la capacidad de seguimiento, depuración y análisis de los logs generados por estos sistemas, lo que resulta fundamental para garantizar un entorno seguro y eficiente para los estudiantes y el personal del instituto.

### **1.2. Alcance del proyecto**

El alcance de este proyecto abarca el diseño e implementación de una solución de centralización y análisis de logs para las aulas de informática del Instituto IES Fernando Aguilar. El objetivo principal es resolver el problema de la falta de una plataforma integrada y eficiente para el almacenamiento, procesamiento y visualización de los logs generados por los diferentes sistemas y el servicio OpenLDAP en este entorno.

### **1.3. Objetivos del proyecto**

El objetivo principal de este proyecto es diseñar e implementar una solución integral que aborde la problemática de la falta de una plataforma integrada y eficiente para el almacenamiento, procesamiento y visualización de los logs generados por los diferentes sistemas y por el servicio OpenLDAP en los talleres de ICO del Instituto IES Fernando Aguilar.

La solución propuesta consiste en recopilar, almacenar, procesar y visualizar los logs de todos los sistemas y del servicio OpenLDAP en una sola plataforma unificada. Para lograr esto, se hará uso de las siguientes herramientas:

- Filebeat
- Docker
- Logstash
- OpenSearch

La solución permitirá centralizar los logs de los sistemas y del servicio OpenLDAP, garantizando su almacenamiento en una única plataforma. Esto facilitará el seguimiento, la depuración y el análisis de los datos, ya que se eliminará la necesidad de consultar múltiples fuentes de logs dispersas.

Además, se utilizará OpenSearch Dashboards, una interfaz de usuario derivada de Kibana, que permitirá visualizar y analizar los logs almacenados en OpenSearch. OpenSearch Dashboards ofrecerá una variedad de gráficos, tablas y otras representaciones visuales para facilitar la comprensión de los datos y ayudar en la detección de patrones o anomalías.

En resumen, este proyecto tiene como objetivo implementar una solución integral que centralice y analice los logs de los sistemas y del servicio OpenLDAP en los talleres de ICO del Instituto IES Fernando Aguilar. La solución se basará en el uso de Filebeat para enviar los logs a Logstash, donde serán procesados, y luego almacenados en OpenSearch. OpenSearch Dashboards se utilizará para la visualización y análisis de los logs almacenados.



## **1.4. Planificación del proyecto**

### **1.4.1. Desglose de tareas.**

Algunas tareas que han sido necesarias realizar para la correcta finalización del proyecto de centralización y análisis de logs son las siguientes:

- **Análisis de requisitos:** Realizar un relevamiento exhaustivo de los requisitos del proyecto, incluyendo los sistemas y servicios involucrados, la cantidad de logs generados, las necesidades de almacenamiento y las funcionalidades requeridas.
- **Investigación de herramientas:** Realizar una investigación detallada sobre las herramientas disponibles, como OpenSearch, Logstash, Filebeat y OpenSearch Dashboards. Evaluar sus características, capacidades, compatibilidad con los sistemas existentes y requisitos de implementación.
- **Diseño de la arquitectura:** Definir la arquitectura de la solución, incluyendo la configuración de los sistemas, la infraestructura necesaria y los flujos de datos entre las diferentes herramientas. Esto implica determinar cómo se enviarán los logs a Logstash, cómo se procesarán y almacenarán en OpenSearch, y cómo se visualizarán en OpenSearch Dashboards.
- **Configuración de Logstash:** Realizar la configuración de Logstash para que pueda recibir, procesar y enriquecer los logs enviados por Filebeat. Esto incluye la definición de pipelines de procesamiento, la configuración de filtros y transformaciones de los datos.
- **Implementación de la plataforma:** Realizar la implementación de la plataforma, incluyendo la instalación y configuración de OpenSearch y OpenSearch Dashboards. Esto implica asegurar la conectividad con Logstash y realizar las configuraciones necesarias para el almacenamiento y búsqueda de los logs.
- **Pruebas y validación:** Realizar pruebas exhaustivas para verificar el correcto funcionamiento de la plataforma. Esto incluye pruebas de envío de logs, procesamiento en tiempo real, almacenamiento y búsqueda de logs, así como la validación de las visualizaciones en OpenSearch Dashboards.

### **1.4.2. Recursos necesarios**

Respecto a los recursos que han sido necesarios para llevar a cabo el proyecto podemos encontrar los siguientes:

- **Evaluar el hardware necesario** para implementar el proyecto, como servidores o máquinas virtuales con suficiente capacidad de almacenamiento y recursos computacionales para ejecutar las aplicaciones y contenedores Docker.
- **Enumerar las herramientas y software necesarios**, como OpenSearch, Logstash, Filebeat y OpenSearch Dashboards. Además, tener en cuenta que se utilizará Docker para facilitar la instalación y gestión de estas herramientas, por lo que se requerirá un entorno de Docker adecuado.
- **Docker:** Asegurarse de contar con una versión de Docker instalada y configurada correctamente en los sistemas donde se desplegarán los contenedores. Además, la necesidad de contar con imágenes de Docker para OpenSearch, Logstash y OpenSearch Dashboards, o bien, realizar su construcción personalizada si fuera necesario.
- **Conectividad:** Comprobar la conectividad de red entre los sistemas donde se ejecutarán los contenedores Docker y cualquier otro sistema o fuente de logs que necesite ser integrado.



#### **1.4.3. Comunicación y seguimiento**

A lo largo del transcurso proyecto se han ido realizando reuniones semanales donde se han mostrado los avances y las posibles alternativas a los caminos elegidos, también se ha dado la resolución de dudas respecto al filtrado de los logs y de la visualización de estos en el panel de OpenSearch Dashboards.



## 2. Capítulo II: Marco Teórico

### 2.1. Syslog & Auth.log

#### 2.1.1. Syslog

En un sistema Linux típico, los registros syslog se almacenan en archivos ubicados en el directorio `/var/log`. Para evitar que estos archivos crezcan indefinidamente y ocupen un espacio excesivo en disco, se utiliza la rotación de archivos.

El proceso de rotación de archivos de registro syslog generalmente es gestionado por una utilidad llamada "logrotate". Logrotate es una herramienta de administración de registros integrada en la mayoría de las distribuciones de Linux y permite programar y configurar la rotación automática de los archivos de registro.

Algunas de las características y opciones comunes de logrotate incluyen:

- **Tamaño máximo:** Se puede configurar un tamaño máximo para los archivos de registro. Una vez que un archivo alcanza ese tamaño, se crea un nuevo archivo y el archivo antiguo se renombra o se comprime.
- **Rotación periódica:** Se puede establecer una programación periódica para la rotación de los archivos de registro. Por ejemplo, se puede configurar la rotación diaria, semanal o mensualmente.
- **Retención de archivos:** Se puede especificar la cantidad de archivos de registro que se mantendrán después de la rotación. Los archivos antiguos se pueden eliminar o almacenar en una ubicación de archivo de respaldo.

Para configurar la rotación de archivos de registro en logrotate, se utilizan archivos de configuración ubicados en el directorio `/etc/logrotate.d/`. Cada archivo de configuración contiene las directivas específicas para la rotación de un archivo de registro en particular.

La rotación de archivos de registro syslog en Linux ayuda a mantener los registros organizados, gestionar el espacio en disco y facilitar la búsqueda y el análisis de eventos. Al programar y configurar adecuadamente la rotación de archivos de registro, se asegura que los registros sean accesibles y utilicen eficientemente los recursos de almacenamiento.

#### 2.1.2. Auth.log

El archivo "auth.log" es uno de los archivos de registro (logs) generados por el sistema operativo Linux. Contiene información relacionada con los eventos de autenticación y autorización en el sistema, incluyendo intentos de inicio de sesión, cambios en la configuración de usuarios, fallas de autenticación y más. Este archivo es especialmente útil para auditar y monitorear la seguridad del sistema.

La rotación de archivos "auth.log" sigue el mismo principio general de la rotación de archivos de registro en Linux.





## 2.2. OpenLDAP

LDAP (Lightweight Directory Access Protocol) o también conocido como “Protocolo Ligero de Acceso a Directorios” es un protocolo de la capa de aplicación TCP/IP que permite el acceso a un servicio de directorio ordenado y distribuido, para buscar cualquier información en un entorno de red. Antes de continuar explicando para qué sirve LDAP, debemos saber qué es un «directorio». Un directorio es un conjunto de objetos con atributos que están organizados de manera lógica y jerárquica, es decir, está en forma de árbol y perfectamente ordenado en función de lo que nosotros queramos, ya sea alfabéticamente, por usuarios, direcciones etc.



Figura 1: Logotipo de OpenLDAP

Generalmente un servidor LDAP se encarga de almacenar información de autenticación, es decir, el usuario y la contraseña, para posteriormente dar acceso a otro protocolo o servicio del sistema. Además de almacenar el nombre de usuario y la contraseña, también puede almacenar otra información como datos de contacto del usuario, ubicación de los recursos de la red local, certificados digitales de los propios usuarios y mucho más. LDAP es un protocolo de acceso que nos permite acceder a los recursos de la red local, sin necesidad de crear los diferentes usuarios en el sistema operativo, además, es mucho más versátil. Por ejemplo, LDAP permite realizar tareas de autenticación y autorización a usuarios de diferentes softwares como Docker, OpenVPN, servidores de archivos como los usados por QNAP, Synology o ASUSTOR entre otros, y muchos más usos.

LDAP puede ser utilizado tanto por un usuario al que se pide unos credenciales de acceso, como también por las aplicaciones para saber si tienen acceso a determinada información del sistema o no. Generalmente un servidor LDAP se encuentra en una red privada, es decir, redes de área local, para autenticar las diferentes aplicaciones y usuarios, pero también podría funcionar sobre redes públicas sin ningún problema.

### 2.2.1. Funcionamiento

LDAP es un protocolo que tiene arquitectura cliente-servidor, por lo tanto, vamos a tener varios clientes que se conectarán a uno o varios servidores LDAP. Generalmente se suele utilizar un solo servidor LDAP donde decenas o cientos de clientes se conectarán a él para acceder a los diferentes recursos de la red local. En el servidor es donde se almacenarán todos los datos relativos al directorio, también se encargará de la autenticación de los usuarios, de comprobar que solamente hay un usuario conectado simultáneamente o varios desde diferentes dispositivos, y de otras tareas que os explicaremos a continuación.

El funcionamiento de LDAP es bastante sencillo, ya que la comunicación es como cualquier otra comunicación entre un cliente y un servidor, tal y como ocurre en Windows con el Directorio Activo.

Las dos acciones básicas que puede hacer un cliente al conectarse son dos, pero antes debemos diferenciar entre autenticación y autorización. La autenticación es el mecanismo por el que nos



identificamos frente a un sistema, por ejemplo, mediante un usuario y contraseña. La autorización es el mecanismo por el cual tenemos o no permiso de hacer algo en el sistema. En un servidor LDAP podemos hacer esto:

- Leer información: para leer la información el cliente debe autenticarse, entonces intentará leer y obtener información del directorio, antes de realizar este paso el servidor se encargará de comprobar si ese usuario en concreto tiene la autorización de leer información.
- Modificar información: para modificar información el proceso es el mismo, pero el servidor comprobará si tenemos permisos de modificación en el servidor.

LDAP también nos permite intercambiar información entre varios servidores, si en un servidor nos autenticamos y éste no tiene la información necesaria, podemos realizar esta consulta a otro servidor que tengamos en la misma red local, para comprobar si efectivamente tenemos esta información o no. Es algo parecido a lo que ocurre con los servidores DNS, que van preguntando uno a otro subiendo por el árbol hasta llegar a los root servers.

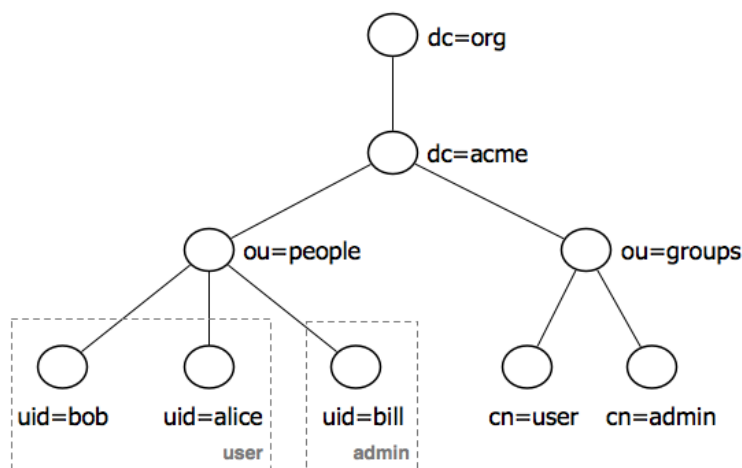


Figura 2: Árbol LDAP.

### 2.2.2. Tipos de operación

En un servidor existen diferentes operaciones que podemos realizar como clientes, a continuación, podéis ver todas las que podemos hacer:

- Add: añadir una nueva entrada. Si la entrada ya existe, el servidor nos lo notificará.
- Modify: modificar una entrada. El protocolo permite tres modificaciones diferentes, añadir nuevo valor, reemplazar valor o eliminar valor.
- Delete: borrar una entrada.
- Search: buscar u obtener entradas del directorio.
- Compare: ver si una entrada nombrada tiene un atributo en concreto.
- Abandon: abortar una petición previa
- Bind: autenticarse en el servidor
- Start TLS: establecer una comunicación segura usando TLS en el protocolo LDAPv3.
- Unbind: cerrar la conexión.



### 2.3. Docker

Docker es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. Docker le permite separar sus aplicaciones de su infraestructura para que puede entregar software rápidamente. Con Docker, puede administrar su infraestructura en de la misma manera que administra sus aplicaciones. Aprovechando el Docker Metodologías para enviar, probar e implementar código rápidamente, puede Reduzca significativamente el retraso entre escribir código y ejecutarlo en producción.



Figura 3: Logotipo de Docker

A continuación, se detallan algunas de las principales ventajas de Docker:

- **Portabilidad:** Los contenedores Docker son ligeros y portátiles, lo que significa que puedes construir una aplicación en un entorno y ejecutarla en otro sin preocuparte por las diferencias de configuración y dependencias. Esto facilita la implementación y el despliegue de aplicaciones en diferentes entornos, ya sea en tu máquina local, en servidores bare-metal, máquinas virtuales o en la nube.
- **Aislamiento:** Docker proporciona un alto nivel de aislamiento para las aplicaciones en contenedores. Cada contenedor tiene su propio entorno de ejecución y recursos aislados, lo que significa que las aplicaciones no afectan ni son afectadas por otras aplicaciones que se ejecutan en el mismo sistema. Esto permite ejecutar aplicaciones de forma segura y evitar conflictos de dependencias.
- **Eficiencia:** Los contenedores Docker son extremadamente eficientes en términos de recursos. A diferencia de las máquinas virtuales tradicionales, que requieren un sistema operativo completo, los contenedores comparten el kernel del sistema operativo host. Esto significa que se pueden ejecutar más contenedores en un mismo servidor físico, lo que maximiza la utilización de recursos y reduce los costos operativos.
- **Escalabilidad:** Docker facilita la escalabilidad de las aplicaciones. Puedes crear imágenes de contenedor que contengan todas las dependencias y configuraciones necesarias para ejecutar una aplicación, y luego instanciar múltiples contenedores a partir de esas imágenes para distribuir la carga de trabajo. Además, Docker proporciona herramientas integradas para la gestión de clústeres de contenedores, lo que facilita la creación y administración de infraestructuras escalables.
- **Despliegue rápido y consistente:** Docker simplifica el proceso de implementación de aplicaciones al encapsular todas las dependencias en un contenedor. Esto significa que puedes desplegar una aplicación Docker en cualquier entorno que tenga Docker instalado, sin tener que preocuparte por las diferencias en la configuración del sistema. Además, al utilizar archivos de configuración (Dockerfile) y versionar las imágenes de contenedor, puedes garantizar que cada instancia de la aplicación se despliegue de manera consistente, lo que facilita la gestión y el mantenimiento a largo plazo.
- **Facilidad de uso:** Docker proporciona una interfaz de línea de comandos (CLI) y una API intuitivas que permiten construir, gestionar y distribuir contenedores de manera sencilla. Además, cuenta con un amplio ecosistema de herramientas y servicios complementarios que simplifican tareas como la orquestación de contenedores, el monitoreo, la gestión de redes y el almacenamiento persistente.



En resumen, Docker ofrece ventajas significativas como portabilidad, aislamiento, eficiencia, escalabilidad, despliegue rápido y consistente, y facilidad de uso. Estas ventajas han convertido a Docker en una herramienta ampliamente utilizada en el desarrollo de aplicaciones modernas y en la implementación de infraestructuras ágiles y flexibles.

## 2.4. OpenSearch

OpenSearch es una plataforma de búsqueda y análisis de código abierto basada en Elasticsearch, que se utiliza para indexar, buscar y analizar grandes volúmenes de datos en tiempo real. OpenSearch es una bifurcación de Elasticsearch y se desarrolla bajo la Fundación OpenSearch, con el objetivo de mantener una versión de código abierto de Elasticsearch y garantizar la transparencia y la comunidad de desarrollo abierta.



Figura 4: Logotipo de OpenSearch

Algunas de las ventajas de OpenSearch:

- **Código abierto:** OpenSearch es una solución de búsqueda y análisis de código abierto, lo que significa que su código fuente está disponible públicamente y puede ser auditado, modificado y mejorado por la comunidad de desarrolladores. Esto brinda transparencia, seguridad y flexibilidad, y permite a los usuarios adaptar y extender la plataforma según sus necesidades.
- **Escalabilidad:** OpenSearch está diseñado para escalar horizontalmente y manejar grandes volúmenes de datos y cargas de trabajo intensivas. Puede distribuir y paralelizar la indexación y la búsqueda en múltiples nodos, lo que permite un rendimiento eficiente y una mayor capacidad de procesamiento a medida que crecen los datos y las demandas de búsqueda.
- **Búsqueda y análisis en tiempo real:** OpenSearch proporciona capacidades de búsqueda y análisis en tiempo real, lo que significa que los datos indexados están disponibles para su búsqueda y análisis casi de inmediato. Esto es especialmente útil para aplicaciones que requieren respuestas rápidas y actualizaciones en tiempo real, como la monitorización de registros, análisis de seguridad, análisis de datos de transmisión, entre otros.
- **Ecosistema y compatibilidad:** OpenSearch cuenta con un ecosistema vibrante y una amplia gama de integraciones y complementos disponibles. Es compatible con una variedad de lenguajes de programación y ofrece bibliotecas y SDKs para facilitar la integración con otras aplicaciones y sistemas. Además, muchos proyectos y herramientas existentes desarrollados para Elasticsearch son compatibles con OpenSearch, lo que facilita la migración de aplicaciones existentes.
- **Funcionalidades avanzadas de búsqueda y análisis:** OpenSearch proporciona una amplia gama de funcionalidades avanzadas para la búsqueda y el análisis de datos, como búsqueda de texto completo, filtrado y consultas complejas, agregaciones, búsqueda geoespacial, análisis de palabras clave, análisis de series temporales y más. Estas funcionalidades permiten a los usuarios realizar consultas sofisticadas y obtener información significativa de los datos indexados.
- **Comunidad y soporte activos:** OpenSearch tiene una comunidad activa de desarrolladores y usuarios que contribuyen al proyecto, brindan soporte, comparten conocimientos y mejoran continuamente la plataforma. Esto garantiza que OpenSearch se mantenga actualizado, se resuelvan problemas y se agreguen nuevas características y mejoras de manera regular.

Elegir entre OpenSearch y Elasticsearch depende de varios factores y requerimientos específicos del proyecto. Es importante tener en cuenta que, si bien OpenSearch puede ser una excelente opción



para muchos casos de uso, Elasticsearch aún es mantenido por Elastic y sigue siendo una opción sólida con soporte comercial y características adicionales en su oferta. Si necesitas soporte empresarial, características específicas de Elastic o estás utilizando productos adicionales de Elastic Stack, es posible que Elasticsearch sea la opción más adecuada.

la elección entre OpenSearch y Elasticsearch dependerá de tus necesidades y preferencias específicas. Se recomienda evaluar cuidadosamente los requerimientos del proyecto, el ecosistema circundante y los objetivos a largo plazo antes de tomar una decisión.

En resumen, OpenSearch ofrece una plataforma de búsqueda y análisis de código abierto, escalable, flexible y con capacidades avanzadas. Su comunidad activa, su compatibilidad con el ecosistema de Elasticsearch y su amplio conjunto de funcionalidades hacen de OpenSearch una opción atractiva para aquellos que buscan una solución de búsqueda y análisis de datos potente y personalizable.

## 2.5. OpenSearch Dashboards

OpenSearch Dashboards es una herramienta de visualización y análisis de datos basada en OpenSearch. Proporciona una interfaz web intuitiva que permite crear y personalizar paneles de control interactivos, gráficos y tablas para visualizar datos indexados en OpenSearch.

Algunos de los componentes clave de OpenSearch Dashboards incluyen los siguientes:

- **Índices:** OpenSearch Dashboards se integra con los índices de OpenSearch, lo que significa que puedes acceder y visualizar los datos almacenados en los índices de OpenSearch. Los índices actúan como repositorios de datos estructurados y son utilizados por OpenSearch para el almacenamiento y recuperación eficiente de datos.
- **Patrones de índices:** Los patrones de índices en OpenSearch Dashboards te permiten definir y aplicar reglas para el nombramiento y creación automática de índices en OpenSearch. Puedes configurar patrones de índices basados en variables de tiempo u otras características, lo que facilita la gestión de datos en evolución y la segmentación de índices según tus necesidades.
- **Dashboards:** OpenSearch Dashboards te permite crear paneles de control personalizados para visualizar tus datos de manera significativa. Puedes agregar diferentes visualizaciones, widgets y métricas en un panel de control para obtener una vista consolidada de tus datos. Los dashboards son altamente personalizables y te permiten organizar y presentar la información de manera visualmente atractiva.
- **Visualizaciones:** OpenSearch Dashboards ofrece una amplia variedad de visualizaciones predefinidas, como gráficos de barras, gráficos circulares, gráficos de líneas, mapas geoespaciales, tablas y más. Estas visualizaciones te permiten explorar y comprender tus datos de manera efectiva. Además, puedes personalizar y combinar estas visualizaciones para adaptarlas a tus necesidades específicas.

Enfocandonos en las ventajas de OpenSearch Dashboards podremos encontrar lo siguiente:

- **Integración con OpenSearch:** OpenSearch Dashboards está diseñado específicamente para trabajar con OpenSearch. Esto significa que aprovecha las funcionalidades y características de búsqueda de OpenSearch, permitiéndote visualizar y analizar los datos.
- **Interfaz intuitiva:** OpenSearch Dashboards proporciona una interfaz web fácil de usar que permite crear y personalizar paneles de control de forma visual, sin necesidad de escribir código. Esto facilita a los usuarios no técnicos la creación y el diseño de visualizaciones y paneles de control interactivos.
- **Amplia variedad de visualizaciones:** OpenSearch Dashboards ofrece una amplia gama de visualizaciones predefinidas, como gráficos de barras, gráficos circulares, gráficos de líneas, mapas geoespaciales, tablas y muchas más. Además, proporciona herramientas para personalizar y combinar estas visualizaciones según tus necesidades.



- Panel de control colaborativo: OpenSearch Dashboards permite compartir paneles de control con otros usuarios y colaborar en tiempo real. Esto facilita la colaboración en proyectos y la visualización de datos compartidos en entornos de equipo o de empresa.

A continuación una comparación de OpenSearch Dashboards con Kibana, otra popular herramienta de visualización y análisis de datos:

- Licencia: OpenSearch Dashboards se distribuye bajo la licencia Apache 2.0, mientras que Kibana se basa en una licencia de código abierto llamada Server Side Public License (SSPL). La licencia Apache 2.0 es considerada más permisiva y amigable para las empresas y proyectos comerciales.
- Compatibilidad: Kibana está diseñado específicamente para trabajar con Elasticsearch, mientras que OpenSearch Dashboards está optimizado para OpenSearch (la bifurcación de Elasticsearch). Si estás utilizando OpenSearch como motor de búsqueda, es recomendable utilizar OpenSearch Dashboards para aprovechar al máximo las funcionalidades y características de OpenSearch.
- Ecosistema y comunidad: Kibana es parte del Elastic Stack, un conjunto de herramientas y productos desarrollados por Elastic, mientras que OpenSearch Dashboards forma parte de la Fundación OpenSearch y se beneficia de una comunidad de desarrollo activa. Ambas herramientas tienen sus propias comunidades y ecosistemas, con diferentes recursos y soporte disponibles.

En resumen, OpenSearch Dashboards es una herramienta de visualización y análisis de datos basada en OpenSearch, diseñada específicamente para trabajar con OpenSearch. Ofrece una interfaz intuitiva, una amplia variedad de visualizaciones y la capacidad de compartir paneles de control de forma colaborativa. Si estás utilizando OpenSearch como motor de búsqueda, OpenSearch Dashboards es una opción sólida para visualizar y analizar tus datos. Sin embargo, si ya estás utilizando Elasticsearch como motor de búsqueda, Kibana es la herramienta recomendada para aprovechar al máximo su ecosistema y características adicionales

## 2.6. Logstash

Logstash es una herramienta de procesamiento de registros (logs) de código abierto desarrollada originalmente por Elastic. Sin embargo, a partir de la versión 7.11 de Elasticsearch, Elastic cambió la licencia de Logstash de Apache 2.0 a la Licencia Server Side Public License (SSPL), que no es una licencia de código abierto reconocida por la Iniciativa de Código Abierto (OSI).

Como resultado, la comunidad de código abierto ha bifurcado el proyecto y ha creado una versión de Logstash compatible con la licencia de código abierto llamada ‘‘OpenSearch Logstash’’ o ‘‘Open Distro for Elasticsearch Logstash’’.

OpenSearch Logstash es una versión del proyecto Logstash adaptada para funcionar con OpenSearch, una bifurcación de código abierto de Elasticsearch. OpenSearch Logstash conserva muchas de las mismas características y funcionalidades que la versión original de Logstash, incluyendo:

- Recopilación de datos: OpenSearch Logstash puede recibir datos de registros de diversas fuentes, como archivos de registro, flujos de eventos en tiempo real, bases de datos y más. Admite una amplia gama de protocolos y formatos de datos para la integración con diferentes fuentes de registros.
- Transformación de datos: OpenSearch Logstash permite realizar transformaciones en los datos de registros antes de enviarlos a OpenSearch. Proporciona una variedad de filtros y capacidades de procesamiento para realizar transformaciones, como filtrado, enriquecimiento y manipulación de datos.
- Canalización de datos: Al igual que en Logstash, OpenSearch Logstash permite construir canalizaciones de datos personalizables utilizando una serie de etapas de procesamiento en





serie. Esto permite diseñar y ajustar las canalizaciones según los requisitos específicos del caso de uso.

- Integración con OpenSearch: OpenSearch Logstash está diseñado para funcionar de manera integrada con OpenSearch. Puede enviar datos de registros procesados a OpenSearch para su almacenamiento, búsqueda y análisis eficientes utilizando las capacidades de búsqueda y análisis de OpenSearch.
- Escalabilidad y disponibilidad: OpenSearch Logstash es escalable y puede ejecutarse en un clúster de nodos, lo que permite una mayor capacidad de procesamiento y tolerancia a fallos en entornos de registros a gran escala.

OpenSearch Logstash es una opción popular para aquellos que desean utilizar Logstash con OpenSearch y mantener una implementación de código abierto sin depender de la versión con licencia SSPL.

## 2.7. Filebeat

Filebeat es otra herramienta de la pila ELK (Elasticsearch, Logstash, Kibana) desarrollada por Elastic. Se utiliza para enviar datos de registros y eventos de archivos de registro a un destino centralizado para su procesamiento y análisis.

Las características principales de Filebeat son las siguientes:

- Recopilación de datos de registros: Filebeat se encarga de leer y recopilar datos de registros de archivos de registro en tiempo real. Puede monitorear y seguir cambios en los archivos de registro, enviando únicamente las actualizaciones o adiciones al destino, lo que minimiza el ancho de banda utilizado y mejora la eficiencia.
- Escalabilidad y ligereza: Filebeat está diseñado para ser ligero y de bajo consumo de recursos, lo que permite su implementación en una amplia gama de entornos y dispositivos. Puede funcionar de manera eficiente incluso en sistemas con recursos limitados.
- Soporte para múltiples formatos de registros: Filebeat es compatible con diversos formatos de registros, incluyendo registros de texto estructurados, archivos de registro JSON y otros formatos personalizados. Esto permite la extracción y envío de información relevante de los archivos de registro, independientemente de su estructura específica.
- Módulos de entrada preconfigurados: Filebeat proporciona módulos de entrada preconfigurados para diversos servicios y aplicaciones populares, como Apache, Nginx, MySQL, PostgreSQL, y muchos más. Estos módulos facilitan la recopilación y envío de datos de registros de forma rápida y sencilla, sin necesidad de realizar configuraciones complicadas.
- Integración con Elasticsearch y otros destinos: Filebeat puede enviar datos de registros a varios destinos, incluyendo Elasticsearch, Logstash, Kafka, entre otros. Esto permite su integración con la pila ELK y otros sistemas de procesamiento y análisis de registros.
- Control y seguridad: Filebeat ofrece características para controlar y asegurar la entrega de los datos de registros. Proporciona mecanismos de reintentos en caso de fallos de conexión y opciones de autenticación y cifrado para garantizar la seguridad de los datos en tránsito.
- Configuración flexible: Filebeat permite una configuración flexible y personalizada según las necesidades del entorno y los requisitos específicos del caso de uso. Se pueden definir múltiples prospectores (prospectors) para monitorear diferentes archivos y ubicaciones, y aplicar filtros para seleccionar y enriquecer los datos antes de enviarlos al destino.

Filebeat es una herramienta ligera y escalable utilizada para recopilar y enviar datos de registros desde archivos de registro a un destino centralizado para su procesamiento y análisis. Con su configuración flexible, soporte para diversos formatos de registros y capacidad de integración con otros componentes de la pila ELK, Filebeat facilita la recopilación eficiente de datos de registros en entornos distribuidos.



### 3. Capítulo III: Análisis y Resultados

En este capítulo se detalla el desarrollo de la solución utilizando las herramientas mencionadas en el [7]Marco Teórico. Gracias a estas seremos capaces de llevar a cabo la centralización y el análisis de los sistemas y de OpenLDAP.

#### 3.1. Análisis de requisitos

##### 3.1.1. OpenSearch & OpenSearch Dashboards & Logstash

Opensearch y Logstash no nos habla sobre los requisitos hardware, sin embargo si nos habla sobre requisitos software como la compatibilidad con el sistema operativo, versiones de java, sistema de archivos y requisitos en red.

La instalación de estas herramientas se harán en el lado del servidor y para ello utilizaremos docker. Docker si nos habla sobre unos requisitos hardware además de los requisitos de software:

- CPU: Se recomienda una CPU de 64 bits compatible con virtualización, como Intel VT o AMD-V.
- Memoria RAM: Docker requiere al menos 2 GB de memoria RAM para su correcto funcionamiento, pero se recomienda disponer de 4 GB o más para un rendimiento óptimo.
- Almacenamiento: Se necesita espacio en disco suficiente para imágenes, contenedores y volúmenes. Asegúrate de contar con suficiente espacio disponible para alojar los componentes de Docker y los datos de tus contenedores.

Si nos fijamos en los requisitos de software necesitaremos lo siguiente:

- Sistema operativo: Docker es compatible con una variedad de sistemas operativos, incluyendo Linux, Windows y macOS. Verifica la documentación oficial de Docker para conocer los requisitos específicos de cada sistema operativo.
- Virtualización: En algunos casos, puede ser necesario habilitar la virtualización en la BIOS del sistema para permitir la ejecución de contenedores de Docker.
- Kernel de Linux: Docker generalmente requiere una versión del kernel 3.10 o superior.

Es muy importante tener buen almacenamiento ya que en el servidor se almacenarán todos los logs de los equipos de las aulas de ICO además de los logs del servicio de OpenLDAP alojado en servus.

##### 3.1.2. Filebeat

Filebeat se trata del cliente encargado de enviar logs a Logstash, este se encontrará instalado en los equipos de las aulas de ICO y también en servus. Al ser un cliente ligero los requisitos hardware no son muy altos:

- CPU: Se recomienda una CPU de 64 bits con al menos 2 núcleos para un rendimiento óptimo, aunque Filebeat puede funcionar en sistemas con menor capacidad de procesamiento.
- Memoria RAM: Filebeat requiere una cantidad mínima de memoria RAM, generalmente alrededor de 2 GB, pero se recomienda disponer de más memoria si se espera un alto volumen de eventos o si se requiere procesamiento adicional de los datos.





## 3.2. Diseño de la solución

### 3.2.1. Arquitectura propuesta para la centralización y procesamiento de logs

Para la centralización y el procesamiento de logs se ha decidido usar una arquitectura compuesta de diferentes etapas. En la siguiente figura observamos como se divide.

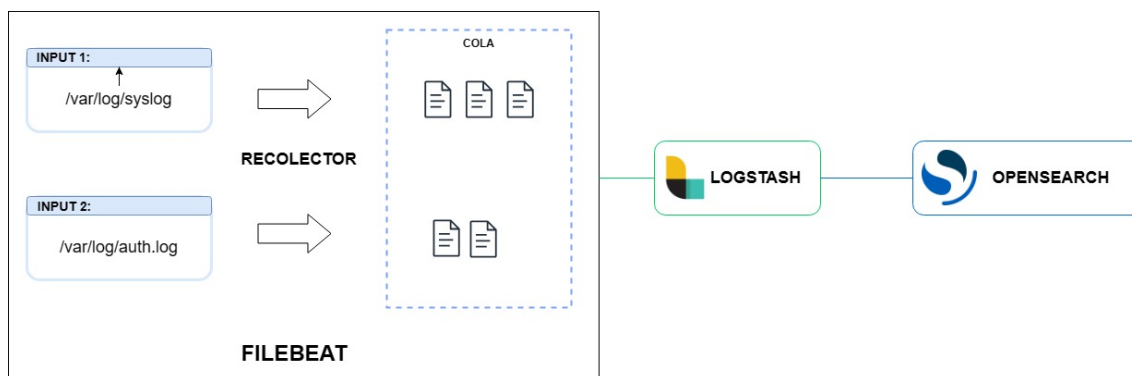


Figura 5: Arquitectura propuesta

El cliente Filebeat instalado en los equipos de ICO recolectará y enviará a logstash dos tipos de logs diferentes, encontraremos syslog y auth.log(en el caso de servus solo enviaremos el syslog). Una vez filebeat los haya recolectado y enviado Logstash se encargará de realizar un filtrado y mapear ciertas variables que nos serán de utilidad más tarde.

Inmediatamente después de haber realizado el filtrado Logstash se encargará de enviar los logs a los nodos de OpenSearch y ya podran ser visualizados desde OpenSearch Dashboards.

### 3.2.2. Configuraciones previas del sistema

Antes de iniciar OpenSearch, se debe revisar algunas configuraciones importantes del sistema que pueden afectar el rendimiento de los servicios.

Deshabilitar la paginación de memoria y el rendimiento de intercambio en el host para mejorar el rendimiento: **sudo swapoff -a**.

Aumentar el número de mapas de memoria disponibles para OpenSearch: **vm.max\_map\_count=262144**

### 3.2.3. Configuración de OpenSearch

Para la creación de los contenedores de OpenSearch se usará un fichero para lanzarlo con docker-compose. Este fichero lo podremos descargar del repositorio de [GitHub](#).

```
1 version: '3'
2 services:
3   opensearch-node1:
4     image: opensearchproject/opensearch:latest
5     container_name: opensearch-node1
6     environment:
7       - cluster.name=opensearch-cluster
8       - node.name=opensearch-node1
9       - discovery.seed_hosts=opensearch-node1,opensearch-node2
10      - cluster.initial_cluster_manager_nodes=opensearch-node1,opensearch-node2
11      - bootstrap.memory_lock=true # along with the memlock settings below,
12      disables swapping
13      - "OPENSEARCH_JAVA_OPTS=-Xms512m -Xmx512m" # minimum and maximum Java heap
14      size, recommend setting both to 50% of system RAM
```



```
13  ulimits:
14      memlock:
15          soft: -1
16          hard: -1
17      nofile:
18          soft: 65536 # maximum number of open files for the OpenSearch user, set to
19                      # at least 65536 on modern systems
20          hard: 65536
21  volumes:
22      - opensearch-data1:/usr/share/opensearch/data
23  ports:
24      - 9200:9200
25      - 9600:9600 # required for Performance Analyzer
26  networks:
27      - opensearch-net
28  opensearch-node2:
29      image: opensearchproject/opensearch:latest
30      container_name: opensearch-node2
31      environment:
32          - cluster.name=opensearch-cluster
33          - node.name=opensearch-node2
34          - discovery.seed_hosts=opensearch-node1,opensearch-node2
35          - cluster.initial_cluster_manager_nodes=opensearch-node1,opensearch-node2
36          - bootstrap.memory_lock=true
37          - "OPENSEARCH_JAVA_OPTS=-Xms512m -Xmx512m"
38      ulimits:
39          memlock:
40              soft: -1
41              hard: -1
42          nofile:
43              soft: 65536
44              hard: 65536
45      volumes:
46          - opensearch-data2:/usr/share/opensearch/data
47      networks:
48          - opensearch-net
49  opensearch-dashboards:
50      image: opensearchproject/opensearch-dashboards:latest
51      container_name: opensearch-dashboards
52      ports:
53          - 5601:5601
54      expose:
55          - "5601"
56      environment:
57          OPENSEARCH_HOSTS: '["https://opensearch-node1:9200","https://opensearch-node2:9200"]'
58      volumes:
59          - opensearch-dashboards:/usr/share/opensearch-dashboards/config
60      networks:
61          - opensearch-net
62  volumes:
63      opensearch-data1:
64      opensearch-data2:
65      opensearch-dashboards:
66  networks:
67      opensearch-net:
```

En este fichero docker-compose crea dos nodos OpenSearch y un nodo OpenSearch Dashboards, también crea la red y los volúmenes donde se mapearan las rutas para poder realizar configuraciones sin tener que entrar a los contenedores.



### 3.2.4. Configuración de Logstash

Esta parte es la más interesante ya que es en la que ocurre toda la magia, es donde llegan los logs y se realiza un filtrado para que podamos visualizarlos de una manera cómoda y eficiente.

Este contenedor lo lanzaremos separado del resto pero estarán todos en la misma red de docker. El fichero para lanzar logstash también lo podemos encontrar en el repositorio de [GitHub](#)

```
1 version: '3'
2 services:
3   logstash:
4     image: opensearchproject/logstash-oss-with-opensearch-output-plugin
5     container_name: logstash
6     ports:
7       - 5044:5044
8     volumes:
9       - /home/pedro/logstash.conf:/usr/share/logstash/pipeline/logstash.conf
10    networks:
11      - pedro_opensearch-net
12
13 networks:
14   pedro_opensearch-net:
```

Destacar que la red y el volumen será a gusto del administrador que vaya a hacer uso de esta herramienta, es decir, puede mapear los volúmenes de docker en el lugar que más cómodo le parezca. Ocurre lo mismo con el nombre de la red.

Ahora hablaremos sobre el fichero logstash.conf. Se trata del fichero donde configuraremos la forma de recibir y enviar logs además del filtrado que se le aplicarán a estos. El fichero se localiza, al igual que los anteriores, en el repositorio de [GitHub](#)

```
1 input {
2   beats {
3     port => 5044
4     codec => plain { charset=>"UTF-8" }
5   }
6 }
7 filter {
8   grok {
9     match => { "message" => "conn=%{NUMBER:conn}" }
10  }
11  grok {
12    match => { "message" => "ACCEPT from IP=%{IPV4:client_ip}" }
13  }
14  if ("BIND" in [message] or "RESULT" in [message] or "ACCEPT" in [message]) and "
15  SEARCH" not in [message] {
16    grok {
17      match => {
18        "message" => [
19          "dn=\"%{DATA:uid},%{GREEDYDATA:dn}\"",
20          "err=%{INT:error_code}",
21          "%{SYSLOGHOST:host}"
22        ]
23      }
24    }
25    kv {
26      source => "dn"
27      field_split => ",",
28      value_split => "=",
29      target => "dn_fields"
30    }
31    ruby {
32      code => "
33        if event.get('client_ip')
34          ip_parts = event.get('client_ip').split('.')
35          taller = 'T' + ip_parts[2]
36          case ip_parts[3]
37            when /\d\d$/
38              taller = 'IC0'
```



```
38         pc = 'DPT0' + ip_parts[3][-1]
39     else
40         pc = case ip_parts[3]
41             when '1'
42                 'SERV1'
43             when /^1\d\d$/
44                 'PC' + ip_parts[3][1..-1]
45             else
46                 nil
47             end
48     end
49     equipo = pc + '-' + taller if pc
50     event.set('PC', pc)
51     event.set('TALLER', taller)
52     event.set('EQUIPO', equipo)
53 end
54 "
55 }
56 aggregate {
57     task_id => "%{conn}"
58     code =>
59         map['bind_result'] ||= '';
60         map['bind_result'] += event.get('message') + ' ';
61         event.to_hash.each { |k,v|
62             map[k] = v unless k == 'message'
63         }
64     "
65     push_previous_map_as_event => true
66     timeout => 5
67 }
68 }
69 if [log][file][path] == "/var/log/auth.log" {
70     if "pam_sss" in [message] {
71         grok {
72             match => { "message" => "user=%{WORD:uid}" }
73         }
74     }
75     ruby {
76         code => "
77             eq = event.get(['host'][hostname])
78             if eq
79                 pc, taller = eq.split('-')
80                 event.set('PC', pc.upcase)
81                 event.set('TALLER', taller.upcase)
82                 event.set('EQUIPO', eq.upcase)
83             end
84         "
85     }
86 }
87 }
88 output {
89     opensearch {
90         hosts => ["https://opensearch-node1:9200", "https://opensearch-node2:9200"]
91         index => "system-logs-%{+YYYY.MM.dd}"
92         user => "admin"
93         password => "admin"
94         ssl => true
95         ssl_certificate_verification => false
96     }
97 }
```



En la configuración se puede ver el uso de los siguientes plugins:

- Beats: Este plugin de entrada permite a Logstash recibir eventos del framework Beats.
- Grok: Analiza texto arbitrario y estructurarlo. Grok es una gran manera de analizar datos de registro no estructurados en algo estructurado y consultable.  
Esta herramienta es perfecta para los registros syslog, apache y otros registros de servidores web, registros mysql, y en general, cualquier formato de registro que se escribe generalmente para los seres humanos y no el consumo de la computadora.
- KV: Este filtro ayuda a analizar automáticamente los mensajes (o campos de evento específicos) que son del tipo clave=valor.  
Esto es ideal para postfix, iptables y otros tipos de registros que tienden a la sintaxis clave=valor.
- Ruby: Este filtro acepta código ruby en línea o un archivo ruby. Las dos opciones son mutuamente excluyentes y tienen formas ligeramente diferentes de funcionar, que se describen a continuación.
- Aggregate: El objetivo de este filtro es agregar la información disponible entre varios eventos (típicamente líneas de registro) pertenecientes a una misma tarea, y finalmente empujar la información agregada en el evento final de la tarea.

Antes de explicar toda la configuración realizada haré un pequeño inciso para explicar como funcionan los logs de OpenLDAP ya que esta configuración se utiliza para estos.

Los logs de OpenLDAP son registros que se generan durante el funcionamiento y la interacción con el servicio de directorio LDAP (Lightweight Directory Access Protocol). Estos registros proporcionan información detallada sobre las operaciones realizadas en el servidor de OpenLDAP, incluyendo acciones de autenticación, consultas, modificaciones y errores.

Cuando el servicio de OpenLDAP se encuentra en funcionamiento, genera eventos y mensajes de registro que se registran en archivos de registro específicos. Los registros pueden variar en nivel de detalle y gravedad, lo que permite realizar un seguimiento exhaustivo de las actividades del servidor y solucionar problemas cuando sea necesario.

Funcionamiento de los logs de OpenLDAP:

- Configuración de los logs: Para habilitar y personalizar los registros de OpenLDAP, es necesario configurar adecuadamente el servidor. Esto se logra a través del archivo de configuración slapd.conf o su equivalente cn=config para implementaciones modernas de OpenLDAP.
- Niveles de registro: OpenLDAP proporciona diferentes niveles de registro que se pueden configurar según las necesidades. Los niveles comunes incluyen:
  - Debug: Proporciona el nivel de detalle más alto para depurar problemas y errores. Es útil durante el desarrollo o la resolución de problemas complejos, pero puede generar una gran cantidad de información y afectar el rendimiento.
  - Stats: Registra estadísticas y métricas relacionadas con el rendimiento y la utilización del servidor.
  - ACL: Registra las decisiones de control de acceso y proporciona información sobre las políticas aplicadas a las operaciones.
  - Trace: Registra el seguimiento de las operaciones de búsqueda, consulta y modificación, incluyendo los filtros utilizados y los resultados obtenidos.
  - Error: Registra errores y mensajes de advertencia relacionados con el servidor LDAP.



■ Principales tipos de logs:

- Log de conexiones (Connection log): Este log registra información sobre las conexiones establecidas con el servidor OpenLDAP, como direcciones IP de origen, identificadores de sesión, protocolos utilizados (LDAP, LDAPS), fechas y horarios de inicio y finalización de las conexiones.
- Log de autenticación (Authentication log): Este log registra información sobre los intentos de autenticación realizados en el servidor, incluyendo los nombres de usuario utilizados, los métodos de autenticación (por ejemplo, Simple Bind, SASL), resultados de autenticación (éxito o fallo) y posibles errores asociados.
- Log de consultas (Query log): Este log registra las operaciones de búsqueda realizadas en el servidor LDAP. Proporciona detalles sobre los filtros utilizados, los atributos consultados, los límites de resultados, los tiempos de respuesta y los resultados devueltos.
- Log de modificaciones (Modification log): Este log registra las operaciones de modificación realizadas en el servidor LDAP, como añadir, eliminar o modificar entradas. Incluye información sobre los cambios realizados, como los atributos afectados y los valores modificados.
- Log de errores (Error log): Este log registra mensajes de error y advertencias generados por el servidor OpenLDAP. Puede incluir detalles sobre errores de conexión, problemas de configuración, fallos en operaciones y otros eventos importantes que requieran atención.
- Log de control de acceso (Access Control log): Este log registra las decisiones tomadas por el servidor OpenLDAP con respecto al control de acceso. Proporciona información sobre las políticas de acceso aplicadas, como las reglas de control de acceso (ACL) que determinan si se permite o deniega el acceso a ciertas operaciones.

Para unir los logs que corresponden a una misma conexión en OpenLDAP, se suele utilizar un identificador de sesión o conexión (session/connection ID) que se registra en los logs. Este identificador único se asigna a cada conexión establecida con el servidor LDAP y se utiliza para agrupar los registros relacionados con esa conexión en particular.

Cuando se establece una conexión con el servidor OpenLDAP, se asigna un ID de sesión o conexión específico a esa conexión. Este ID se registra en los logs junto con otros detalles de la conexión, como la dirección IP del cliente, el nombre de usuario y el protocolo utilizado.

Para realizar el seguimiento de los logs correspondientes a una misma conexión, se puede buscar o filtrar los registros en función del ID de sesión o conexión. Esto permite identificar todos los registros asociados a una conexión específica y analizarlos de manera conjunta para comprender mejor el flujo de eventos y las acciones realizadas durante esa conexión en particular.

Retomando la configuración de logstash, en las **líneas ocho a trece** hago uso del plugin grok para capturar el id de la conexión de OpenLDAP y también capturar la dirección ip del host remoto en el cual el usuario se autentifica.

En las **líneas catorce** a la **veintitres** mapeamos otros campos como el uid del usuario, el dn(nombre distinguido, es una serie de pares clave/valor separados por comas que se utilizan para identificar las entradas de forma exclusiva en la jerarquía de directorios LDAP), además de otros campos como el código de error.

Haciendo uso del plugin **KV** en las **líneas venticuatro** a la **veintinueve** separaremos el dn para poder sacar el ciclo al que corresponde o si es profesor(DAW,ASIR,SMR,DAM,PROFESORES)

Gracias a **ruby** podremos crear variables interesantes como las siguientes:

- Equipo: Se trata del nombre del equipo al completo, este sale de la suma del nombre del pc más el taller, por ejemplo, PC01-T2.



- Taller: Se trata del taller que corresponde. Cada aula ICO tiene un nombre asignado, en el caso del ejemplo anterior el PC01 se localiza en el taller dos que es el aula ICO2.
- Pc: Es el nombre que tiene asignado cada equipo de cada aula, todos se repiten por eso necesitan el nombre completo.

También en el código de ruby tenemos en cuenta los equipos del profesor y del departamento ya que estos reciben un nombre distinto. Un ejemplo para la explicación del código puede ser la siguiente:

Tenemos las direcciones 10.4.2.1, 10.4.1.101, 10.4.2.21. Dividimos las direcciones de la siguiente forma para que quede de la siguiente forma: 10.4.**taller.pc**

Una vez dividida tenemos que comprobar el **taller**, dependiendo del número será T1,T2,T3 o T4 pero también existe el departamento y este recibe el número de **taller dos**. Para comprobar que pertenece al departamento y no al taller de ICO dos debemos fijarnos en la parte del **pc**. Si en esta parte encontramos un **número de dos cifras** corresponde al departamento.

Una vez solucionado este problema, nos queda comprobar si se trata del equipo del profesor o de los equipos de los alumnos, es muy sencillo, si la parte de **pc** tiene una cifra y esa es el número uno se trata del equipo del profesor y recibirá el nombre **SERV1-<TALLER>**. Si es un número de tres cifras corresponde a un equipo de los alumnos.

Entonces las direcciones ips quedarían de la siguiente forma:

- 10.4.2.1:
  - EQUIPO: SERV1-T2
  - TALLER: T2
  - PC: SERV1
- 10.4.1.101:
  - EQUIPO: PC01-T1
  - TALLER: T1
  - PC: PC01
- 10.4.2.21:
  - EQUIPO: DEPT1-INF
  - TALLER: INF
  - PC: DEPT1

Siguiendo con las **líneas cincuenta y seis** a la **setenta y cuatro** encontramos el plugin **aggregate** con el que lograremos unir los eventos pertenecientes a la misma conexión de ldap haciendo uso de la variable **conn** que mapeamos anteriormente con **grok**.

Finalmente, con las **líneas setenta y cinco** a la **noventa y tres**, relacionaremos los logs de ldap con los distintos los de los sistemas, más concretamente con los logs de auth.log. Esto lo conseguiremos gracias al módulo pam\_sss.so. Se utiliza en sistemas Linux para integrar el protocolo LDAP (Lightweight Directory Access Protocol) con el sistema de autenticación basado en PAM. En particular, pam\_sss.so permite la autenticación de usuarios utilizando información almacenada en un servidor LDAP a través del servicio SSSD (System Security Services Daemon).

En estas líneas también será necesario mapear el nombre de los equipos pero es más sencillo ya que este lo sacaremos de la variable host.hostname que se mapea de forma automática al enviar los logs.



### 3.2.5. Configuración de Filebeat

Para hacer cambios de configuración en Filebeat debemos dirigirnos al archivo de configuración localizado en `/etc/filebeat/filebeat.yml` en el cual estableceremos lo siguiente:

```
1 filebeat.config.modules:
2   path: ${path.config}/modules.d/*.yaml
3   reload.enabled: false
4
5 setup.template.settings:
6   index.number_of_shards: 1
7
8 output.logstash:
9   hosts: ["<direccion>:<puerto>"]
10
11 processors:
12   - add_host_metadata:
13     when.not.contains.tags: forwarded
14   - add_cloud_metadata: ~
15   - add_docker_metadata: ~
16   - add_kubernetes_metadata: ~
```

A continuación, antes de reiniciar el servicio, activaremos el módulo `system`.

Este módulo recopila y analiza los registros creados por el servicio de registro del sistema de las distribuciones comunes basadas en Unix/Linux.

Para activarlo haremos uso de la siguiente instrucción:

```
1 #Activar el modulo system
2 filebeat modules enable system
3 #Comprobar los modulos activados
4 filebeat modules list
```

Ya activado debemos de modificar el fichero correspondiente que se encontrará en `/etc/filebeat/modules.d/system.yml`

```
1 # Module: system Docs: https://www.elastic.co/guide/en/beats/filebeat/master/
   filebeat-module-system.html
2
3 - module: system
4   # Syslog
5   syslog:
6     enabled: true
7     var.paths: ["/var/log/syslog"]
8
9   auth:
10    enabled: true
11    var.paths: ["/var/log/auth.log"]
```

Como ya señalé anteriormente `servus` tendrá desactivado el envío de los logs que corresponden al fichero `auth.log`.

Una vez realizadas estas configuraciones ya podremos reiniciar el servicio.



## CFGS 2º ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS Y REDES

Centralación y análisis de logs mediante Opensearch



### 3.2.6. Configuración de OpenSearch Dashboards

Una vez configurado Logstash y Filebeat es hora de realizar la configuración en la fase final, es decir, la visualización.

Accederemos al dashboard haciendo uso de la dirección ip del servidor que aloja los contenedores y el puerto 5601. Cuando estemos dentro nos dirigiremos al apartado **Index Management** y luego a **Índices** aquí podremos comprobar que el cliente de filebeat y logstash estan funcionando correctamente ya que podremos ver que llegan los logs de los sistemas.

Index	Health	Managed by policy	Status	Total size	Size of primaries	Total documents	Deleted documents	Primaries	Replicas
<input type="checkbox"/> system-logs-2023.06.18	Green	No	Open	39.1mb	39.1mb	119816	0	1	1
<input type="checkbox"/> security-auditlog-2023.06.18	Green	No	Open	491.3kb	491.3kb	165	0	1	1
<input type="checkbox"/> security-auditlog-2023.06.13	Green	No	Open	780.8kb	780.8kb	766	0	1	1
<input type="checkbox"/> .opensearch-observability	Green	No	Open	6kb	6kb	1	0	1	1
<input type="checkbox"/> .opensearch-notifications-config	Green	No	Open	208b	208b	0	0	1	1
<input type="checkbox"/> .opendistro_security	Green	No	Open	71.7kb	71.7kb	10	0	1	1
<input type="checkbox"/> .opendistro-reports-instances	Green	No	Open	6.1kb	6.1kb	1	0	1	1

Figura 6: OpenSearch: Índices

En la Figura 6 se puede ver el índice **system-logs-2023.06.18** que corresponde al que Logstash envía. Ya comprobado el índice debemos de crear un **patrón de índice** el cual nos ayudará a recuperar los datos. Para ello nos dirigimos a **Stack Management > Index Patterns > Create index pattern**

Create index pattern

An index pattern can match a single source, for example, `filebeat-4-3-22`, or multiple data sources, `filebeat-*`.

Read documentation

Step 1 of 2: Define an index pattern

Index pattern name

`system*`

Use an asterisk (\*) to match multiple indices. Spaces and the characters `\`, `/`, `.`, `<`, `>`, `[`, `]` are not allowed.

☐ Include system and hidden indices

✓ Your index pattern matches 1 source.

`system-logs-2023.06.18`

Index

Rows per page: 10

Figura 7: OpenSearch: Index Pattern 2

# CFGS 2º ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS Y REDES

## Centralación y análisis de logs mediante Opensearch

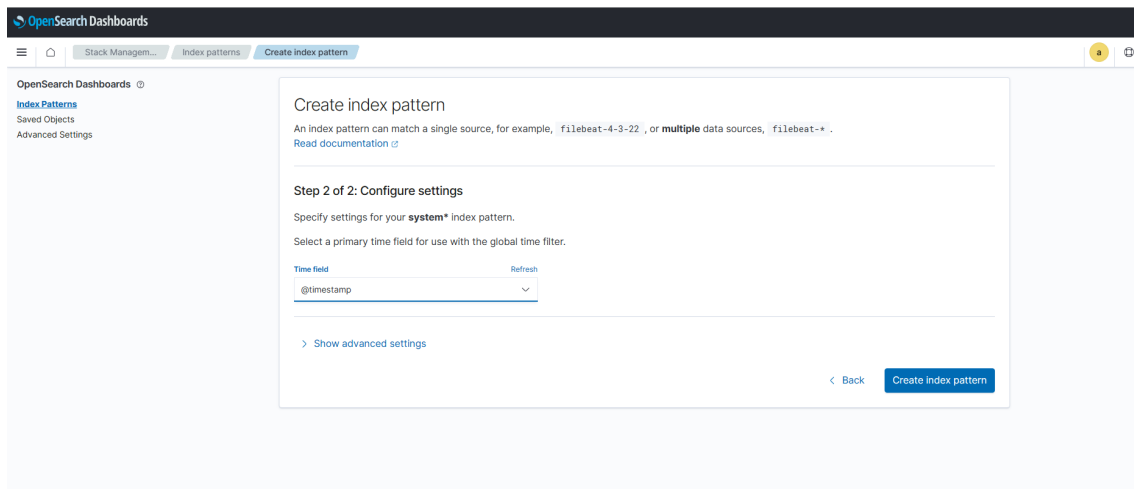


Figura 8: OpenSearch: Index Pattern 2

Cuando terminemos de crearlo ya podremos crear nuestro Dashboard, pero antes podemos comprobar los logs recibidos en la herramienta **Discover**.

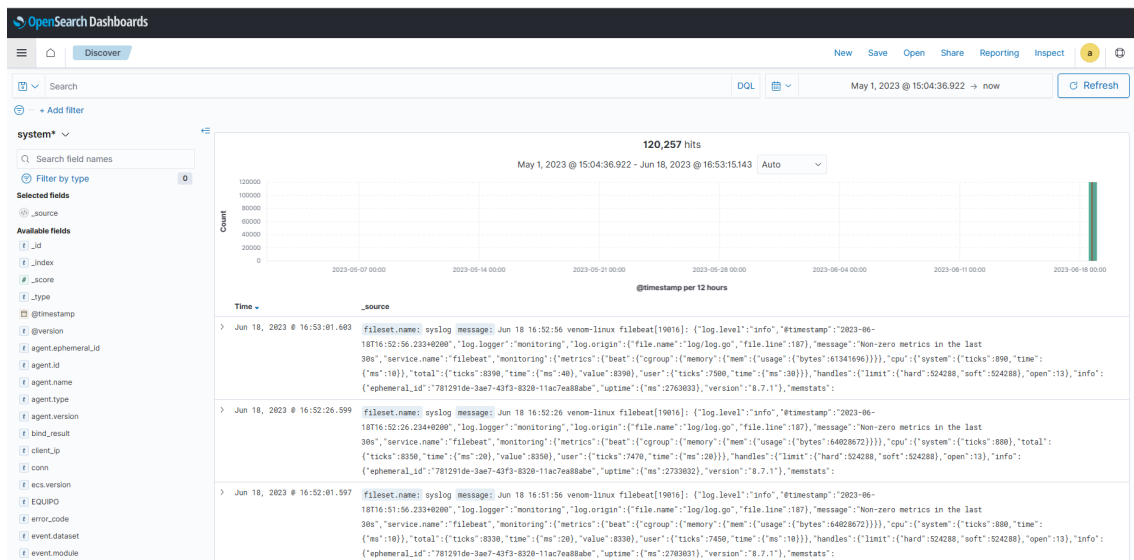


Figura 9: OpenSearch: Discover

Por último, para crear nuestro primer dashboard nos dirigimos a la herramienta **Dashboard** y pulsamos sobre el botón **Create new dashboard** y tendremos una vista en la cual podremos agregar las diferentes visualizaciones.

## CFGS 2º ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS Y REDES

Centralación y análisis de logs mediante Opensearch

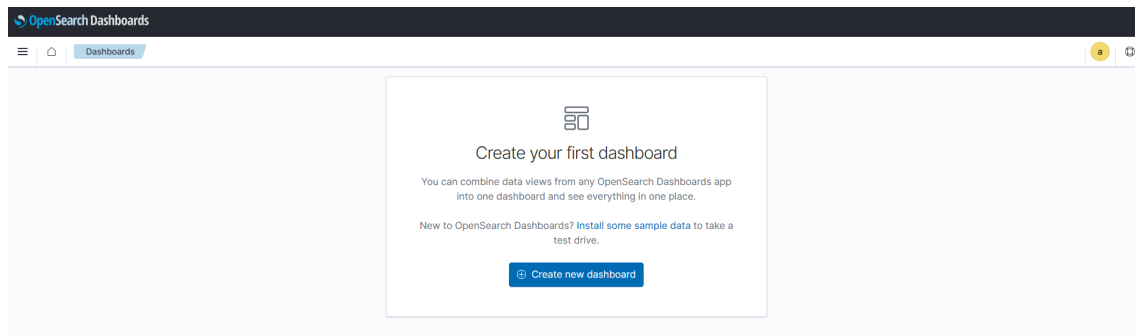


Figura 10: OpenSearch: Dashboard

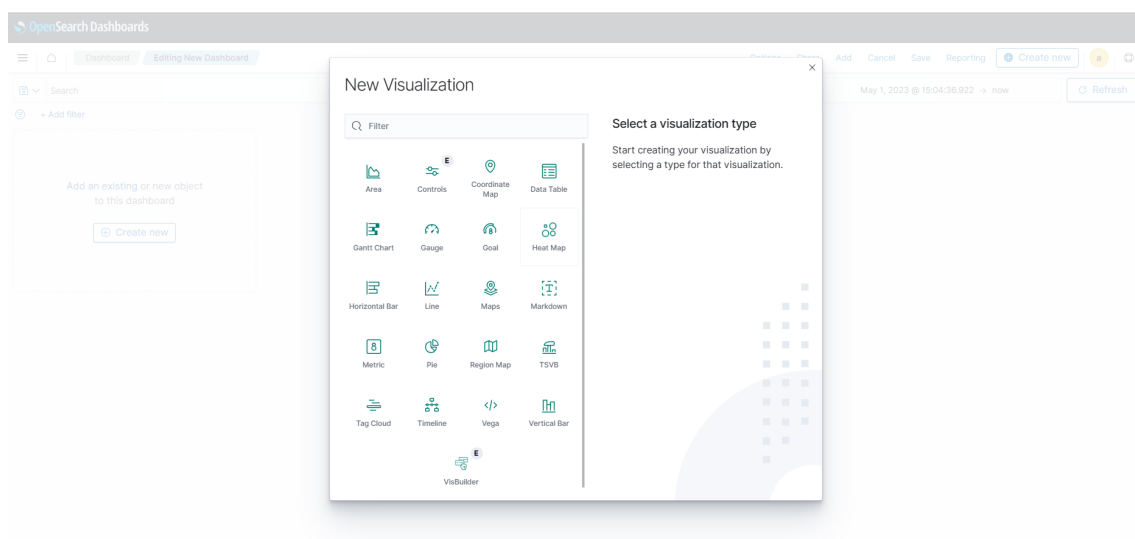


Figura 11: OpenSearch: Visualizaciones



### 3.3. Implementación

#### 3.3.1. Descripción de las etapas y pasos seguidos durante la implementación de la solución.

Durante la implementación de la solución, se siguieron varias etapas y se llevaron a cabo distintos pasos para abordar la problemática de almacenamiento, procesamiento y visualización de los logs en los talleres de ICO del Instituto IES Fernando Aguilar. A continuación, se detalla cada una de estas etapas y los pasos seguidos:

- **Análisis de requisitos:** Se realizó un exhaustivo análisis de los requisitos para comprender las necesidades de almacenamiento, procesamiento y visualización de los logs generados por los diferentes sistemas y por el servicio OpenLDAP. Se tomaron en cuenta factores como la capacidad de almacenamiento, la escalabilidad, la seguridad y la eficiencia.
- **Diseño de la arquitectura:** Se diseñó la arquitectura de la solución, considerando las herramientas propuestas, como Filebeat, Docker, Logstash, OpenSearch y OpenSearch Dashboards. Se definieron los componentes, su interconexión y los flujos de datos necesarios para centralizar y analizar los logs de manera eficiente.
- **Evaluación de opciones de clientes para el envío de logs:** Se investigaron diferentes opciones de clientes para enviar los logs a Logstash. Se realizaron comparativas entre herramientas como rsyslog, fluent-bit y filebeat, considerando aspectos como facilidad de configuración, soporte de formatos de logs y capacidad de transporte seguro.
- **Configuración de las herramientas seleccionadas:** Se procedió a la configuración de las herramientas seleccionadas. Se establecieron los parámetros necesarios en Filebeat para la recopilación y envío de los logs, se configuró Docker para el despliegue de los contenedores, se establecieron los filtros y transformaciones en Logstash para el procesamiento de los logs, y se configuró OpenSearch y OpenSearch Dashboards para el almacenamiento y visualización de los logs.
- **Configuración de las herramientas seleccionadas:** Se procedió a la configuración de las herramientas seleccionadas. Se establecieron los parámetros necesarios en Filebeat para la recopilación y envío de los logs, se configuró Docker para el despliegue de los contenedores, se establecieron los filtros y transformaciones en Logstash para el procesamiento de los logs, y se configuró OpenSearch y OpenSearch Dashboards para el almacenamiento y visualización de los logs.
- **Pruebas y validación:** Se llevaron a cabo pruebas exhaustivas para verificar el correcto funcionamiento de la solución implementada. Se probaron diferentes casos de uso y se verificó que los logs se estuvieran recopilando, almacenando, procesando y visualizando correctamente en OpenSearch Dashboards. Se realizaron ajustes y correcciones según los resultados obtenidos.
- **Documentación y capacitación:** Se elaboró una documentación completa de la implementación de la solución, incluyendo manuales de uso y administración. Además, se impartió capacitación al personal encargado de administrar la solución y se brindó soporte para asegurar un correcto manejo y mantenimiento de la plataforma.

A través de estas etapas y pasos, se logró implementar una solución integral que centraliza y analiza los logs de los sistemas y del servicio OpenLDAP en los talleres de ICO del Instituto IES Fernando Aguilar, brindando una plataforma eficiente y unificada para su almacenamiento, procesamiento y visualización.



### **3.3.2. Mención de posibles desafíos o problemas encontrados durante la implementación y cómo se resolvieron.**

Durante el proceso de implementación, surgieron varios desafíos y problemas, especialmente relacionados con el filtrado de los logs utilizando la herramienta Logstash. Se realizaron pruebas exhaustivas utilizando diferentes configuraciones y plugins con el objetivo de unir y relacionar los logs de LDAP.

Uno de los desafíos específicos fue utilizar el plugin multiline para unir los logs de LDAP que estaban dispersos en múltiples líneas. Sin embargo, a pesar de varios intentos, no se logró un resultado exitoso utilizando esta técnica.

Para abordar este problema, se exploraron diferentes enfoques alternativos. Se decidió utilizar el plugin “aggregate” de Logstash, que permitía agrupar los eventos de logs basados en un campo específico y realizar operaciones personalizadas sobre ellos. Se configuró el plugin “aggregate” para unir los logs de LDAP relacionados en un solo evento, utilizando identificadores de conexión como referencia para agruparlos correctamente.

Esta solución basada en el plugin “aggregate” demostró ser efectiva y permitió reunir y relacionar correctamente los logs de LDAP. Se realizaron pruebas exhaustivas para garantizar la precisión y consistencia de los resultados.

Además de los desafíos relacionados con el filtrado y la unión de los logs de LDAP, se enfrentaron otros problemas menores durante la implementación. Estos incluyeron ajustes de configuración, solución de errores de conectividad, y garantizar la compatibilidad entre las diferentes versiones de las herramientas utilizadas.

Cada problema identificado fue abordado de manera sistemática, analizando cuidadosamente las causas subyacentes y aplicando soluciones adecuadas. Se realizaron consultas a la documentación oficial, se buscó ayuda en foros y comunidades de usuarios, y se llevaron a cabo pruebas rigurosas para validar las soluciones implementadas.

En resumen, los desafíos encontrados durante la implementación se enfrentaron de manera proactiva y se buscaron soluciones efectivas. Esto permitió superar los obstáculos y lograr una solución integral que cumpliera con los requisitos de almacenamiento, procesamiento y visualización de los logs en los talleres de ICO.



## Centralización y análisis de logs mediante Opensearch



Después:

[illegible]

30



## 4. Capítulo IV: Conclusiones y Líneas Futuras

Después de llevar a cabo la implementación de la solución integral para el almacenamiento, procesamiento y visualización de los logs en los talleres de ICO del Instituto IES Fernando Aguilar, se pudieron observar una serie de resultados significativos.

En cuanto a los resultados obtenidos, se pudo ver que la centralización de los logs en una plataforma unificada ha permitido mejorar la eficiencia en el seguimiento, depuración y análisis de los datos generados por los diferentes sistemas y el servicio OpenLDAP. La eliminación de la necesidad de consultar múltiples fuentes de logs dispersas ha facilitado la tarea de comprensión y detección de patrones o anomalías en los registros.

La combinación de Filebeat, Logstash, OpenSearch y OpenSearch Dashboards ha resultado ser una solución sólida y eficiente para el procesamiento, almacenamiento y visualización de logs. Mediante Filebeat, se logra enviar los logs a Logstash, donde son procesados y posteriormente almacenados en OpenSearch. A su vez, OpenSearch Dashboards proporciona una interfaz intuitiva que facilita la creación y personalización de paneles de control interactivos, gráficos y tablas, brindando a los usuarios una experiencia amigable y accesible.

En cuanto a las líneas futuras de trabajo, se plantea continuar mejorando y ajustando la configuración de los filtros y patrones de indexación para optimizar la captura y visualización de los logs de forma más precisa. Asimismo, se explorará la posibilidad de integrar alertas y notificaciones para detectar y responder rápidamente a eventos o patrones anómalos en los logs.

También se podría investigar la integración de OpenSearch con otros sistemas y servicios relacionados, como la mejora de la detección de amenazas en sistemas de seguridad perimetral o la integración con herramientas de análisis de big data para obtener una visión más completa de los patrones de comportamiento.

Otra línea de investigación sería el desarrollo de técnicas de monitoreo y alertas en tiempo real más avanzadas. Esto implica configurar alertas basadas en eventos específicos en los logs y explorar opciones para la detección automática de anomalías o patrones inusuales de comportamiento.

Por último, sería interesante investigar en el campo del análisis avanzado de logs utilizando técnicas como el aprendizaje automático y la inteligencia artificial. Esto implica explorar algoritmos y modelos para la detección de patrones, el análisis de comportamiento y la correlación de eventos, lo que podría revelar conocimientos más profundos y valiosos sobre los datos de logs almacenados en OpenSearch.

En conclusión, la implementación de la solución ha demostrado ser exitosa en la centralización y análisis de los logs en los talleres de ICO. Las líneas futuras de trabajo permitirán seguir optimizando y mejorando la solución para satisfacer las necesidades cambiantes de los talleres y garantizar un seguimiento efectivo de los sistemas y servicios.





## Referencias

- [1] **Installing OpenSearch**, [En línea]. Disponible en: <https://opensearch.org/docs/latest/install-and-configure/install-opensearch/index/>. [Accedido: fecha]
- [2] **Docker**, [En línea]. Disponible en: <https://opensearch.org/docs/latest/install-and-configure/install-opensearch/docker/>. [Accedido: fecha]
- [3] **Logstash**, [En línea]. Disponible en: <https://opensearch.org/docs/latest/tools/logstash/index/>. [Accedido: fecha]
- [4] **Logstash: Recopila, parsea y transforma logs | Elastic**, [En línea]. Disponible en: <https://www.elastic.co/es/logstash/>. [Accedido: fecha]
- [5] **Input plugins**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/input-plugins.html>. [Accedido: fecha]
- [6] **Output plugins**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/output-plugins.html>. [Accedido: fecha]
- [7] **Filter plugins**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/filter-plugins.html>. [Accedido: fecha]
- [8] **Codec plugins**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/codec-plugins.html>. [Accedido: fecha]
- [9] **Multiline codec plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-codecs-multiline.html>. [Accedido: fecha]
- [10] **Beats input plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-inputs-beats.html>. [Accedido: fecha]
- [11] **Tcp input plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-inputs-tcp.html>. [Accedido: fecha]
- [12] **File input plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-inputs-file.html>. [Accedido: fecha]
- [13] **Aggregate filter plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-filters-aggregate.html>. [Accedido: fecha]
- [14] **Kv filter plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-filters-kv.html>. [Accedido: fecha]
- [15] **Ruby filter plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-filters-ruby.html>. [Accedido: fecha]
- [16] **Grok filter plugin**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html>. [Accedido: fecha]
- [17] **Analyze your OpenLDAP Logs**, [En línea]. Disponible en: <https://www.neteye-blog.com/2018/03/analyze-your-openldap-logs/>. [Accedido: fecha]
- [18] **¿Qué es Grok y como usarlo?**, [En línea]. Disponible en: <https://alexmarket.medium.com/qu%C3%A9-es-grok-y-como-usarlo-68729a3dd41f>. [Accedido: fecha]
- [19] **OpenSearch Dashboards**, [En línea]. Disponible en: <https://opensearch.org/docs/latest/dashboards/index/>. [Accedido: fecha]
- [20] **Kibana: Explora, visualiza y descubre datos | Elastic**, [En línea]. Disponible en: <https://www.elastic.co/es/kibana/>. [Accedido: fecha]
- [21] **Filebeat: Análisis de logs ligero y Elasticsearch**, [En línea]. Disponible en: <https://www.elastic.co/es/beats/filebeat>. [Accedido: fecha]



- [22] **Filebeat reference**, [En línea]. Disponible en: <https://www.elastic.co/guide/en/beats/filebeat/current/index.html>. [Accedido: fecha]
- [23] **An End to End Observability Pipeline**, [En línea]. Disponible en: <https://fluentbit.io/>. [Accedido: fecha]
- [24] **Fluent Bit v2.1 Documentationl**, [En línea]. Disponible en: <https://docs.fluentbit.io/manual>. [Accedido: fecha]
- [25] **Ruby**, [En línea]. Disponible en: <https://www.ruby-lang.org/es/documentation/>. [Accedido: fecha]
- [26] **Lenguaje de programación Ruby: características y utilidades**, [En línea]. Disponible en: <https://www.mytaskpanel.com/lenguaje-de-programacion-ruby/>. [Accedido: fecha]
- [27] **Ruby en 20 minutos**, [En línea]. Disponible en: <https://www.ruby-lang.org/es/documentation/quickstart/>. [Accedido: fecha]
- [28] **Develop faster. Run anywhere.**, [En línea]. Disponible en: <https://www.docker.com/>. [Accedido: fecha]
- [29] **Terms list might be incomplete because the request is taking too long**, [En línea]. Disponible en: <https://forum.opensearch.org/t/terms-list-might-be-incomplete-because-the-request-is-taking-too-long/9034>. [Accedido: fecha]
- [30] **pam\_sss(8) - Linux man page**, [En línea]. Disponible en: [https://linux.die.net/man/8/pam\\_sss](https://linux.die.net/man/8/pam_sss). [Accedido: fecha]



## Anexo A: Manual de instalación.

### OpenSearch requisitos (INSTALACIÓN NORMAL).

#### Compatibilidad del sistema operativo y del sistema de archivos.

Se recomienda instalar OpenSearch en Red Hat Enterprise Linux (RHEL) o distribuciones Linux basadas en Debian que utilicen systemd. Evitar utilizar un sistema de archivos de red para el almacenamiento de nodos en un flujo de trabajo de producción. El uso de un sistema de archivos de red para el almacenamiento de nodos puede causar problemas de rendimiento en el clúster debido a factores como las condiciones de la red (como la latencia o el rendimiento limitado) o las velocidades de lectura y escritura. Se debe utilizar unidades de estado sólido (SSD) instaladas en el host para el almacenamiento de nodos siempre que sea posible.

#### Compatibilidad con Java.

Versión de OpenSearch	Versiones de Java compatibles	Versión Java incluida
1.0 - 1.2.x	11, 15	15.0.1+9
1.3.x	8, 11, 14	8, 11, 14
2.0.0	2.0.0	17.0.2+8

Para utilizar una instalación de Java es necesario establecer la siguiente variable de entorno:  
**OPENSEARCH\_JAVA\_HOME**

#### Requisitos de red.

Los siguientes puertos deben estar abiertos para los componentes de OpenSearch.

Número de puerto	Componente OpenSearch
443	Paneles de OpenSearch en AWS OpenSearch Service con cifrado en tránsito (TLS)
5601	Paneles de OpenSearch
9200	OpenSearch REST API
9250	Búsqueda entre clústeres
9250	Búsqueda entre clústeres
9600	Analizador de rendimiento

#### Ajustes importantes.

Para cargas de trabajo de producción, es necesario aumentar el valor de la propiedad del kernel **vm.max\_map\_count** se trata de una propiedad del kernel utilizada para definir el número máximo de áreas de mapa de memoria que un proceso puede tener. Debe recibir el valor **262144**.



## OpenSearch requisitos (INSTALACIÓN CON DOCKER).

Para empezar necesitaremos descargar e instalar Docker en nuestro sistema, para ello podemos seguir los pasos de la [guía oficial](#).

### Configuración importante del host.

Antes de iniciar OpenSearch, se debe revisar algunas configuraciones importantes[4] del sistema que pueden afectar el rendimiento de los servicios.

Deshabilitar la paginación de memoria y el rendimiento de intercambio en el host para mejorar el rendimiento: **sudo swapoff -a**.

Aumentar el número de mapas de memoria disponibles para OpenSearch: **vm.max\_map\_count=262144**

### Despliegue de los contenedores.

Para desplegar los contenedores se debe descargar el fichero docker-compose.yml alojado en [GitHub](#).

Antes de lanzar estos contenedores con la instrucción docker-compose es muy probable que interese modificarlos para mapear los volúmenes en la ruta más adecuada. También es interesante cambiar el nombre de la red(no es obligatorio pero es recomendable elegir un propio.)

Una vez desplegados los contenedores es importante modificar ciertos parámetros del Dashboard para que este funcione correctamente.

Haciendo uso del editor nano o vi modificaremos el archivo **opensearch\_dashboards.yml**, situado en la ruta donde se encuentre el volumen correspondiente, y establecemos las siguientes variables:

- **opensearchDashboards.autocompleteTimeout=1000000**
- **opensearchDashboards.autocompleteTerminateAfter: 1000000**

Con esto solucionaremos un problema a la hora de intentar utilizar los controles que si no añadimos esto cuando intentemos filtrar nos aparecerá el siguiente error:

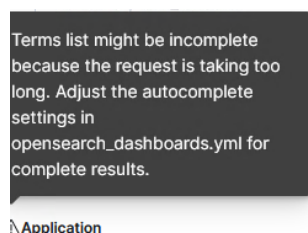


Figura 14: Anexo A: Error dashboard

Una vez desplegado el todos los contenedores haciendo uso de **docker ps** deberíamos ver 4 contenedores:



```
opensearch-5$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
17c84e14750	opensearchproject/logstash-oss-with-opensearch-output-plugin:8.6.1	"/usr/local/bin/dock..."	3 days ago	Up 3 minutes	0.0.0.0:5044->5044/tcp, :::5044->5044/tcp, 9600/tcp
18057183af80	opensearchproject/opensearch-dashboards:latest	"/opensearch-dashbo..."	3 days ago	Up 3 minutes	0.0.0.0:5601->5601/tcp, :::5601->5601/tcp
18774a523cd5	opensearchproject/opensearch:latest	"/opensearch-docker..."	3 days ago	Up 3 minutes	9200/tcp, 9300/tcp, 9600/tcp, 9650/tcp
c9c925bd4874	opensearchproject/opensearch:latest	"/opensearch-docker..."	3 days ago	Up 3 minutes	0.0.0.0:9200->9200/tcp, :::9200->9200/tcp, 9300/tcp, 0.0.0.0:9600->9600/tcp, :::9600->9600/tcp, 9650/tcp

```
opensearch-5$
```

Figura 15: Anexo A: Comprobación de contenedores

Ya podríamos acceder al panel a través de la url: `http://<ip>:5601`

## Instalación y configuración de logstash

Para la instalación de logstash haremos uso de dos ficheros alojados en [GitHub](#), `logstash.conf` y `docker-compose-logstash.yml`.

El archivo `logstash.conf` corresponde a la configuración principal y debe ser alojado en el lugar que el administrador desee pero luego hay que indicar la ruta en el fichero `docker-compose-logstash.yml`.

También es muy importante modificar el fichero de `docker-compose` para indicar la red ya que logstash debe estar en la misma que los contenedores de OpenSearch.

## Instalación de los clientes.

Para enviar los logs de los equipos clientes hacia Logstash usaré **Filebeat**, se trata de un agente ligero que nos permite enviar y centralizar logs y archivos.

### Instalación de Filebeat.

#### 1.Descargar e instalar la clave de firma pública.

```
1 wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

#### 2.Instalar el paquete apt-transport-https antes de continuar(Debian).

```
1 sudo apt-get install apt-transport-https
```

#### 3.Guardar la definición del repositorio en `/etc/apt/sources.list.d/elastic-8.x.list`

```
1 echo "deb https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-8.x.list
```

#### 4.Ejecutar apt update e instalar.

```
1 apt update && apt install filebeat
2 systemctl enable filebeat
```



## Fichero filebeat.yml

En este fichero debemos de buscar el apartado **Logstash Output** y definiremos la variable **hosts** con el valor que corresponda, es decir, la dirección ip de la máquina que contiene el plugin Logstash y el puerto que le hayamos establecido.

```
#password: changeme

# ----- Logstash Output -----
output.logstash:
  # The Logstash hosts
  hosts: ["192.168.122.2:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

# ----- Processors -----
```

Figura 16: Anexo A: Fichero filebeat.yml

Además debemos de tener establecido lo siguiente:

```
1 filebeat.config.modules:
2   path: ${path.config}/modules.d/*.yaml
3   reload.enabled: false
4
5 setup.template.settings:
6   index.number_of_shards: 1
7
8 output.logstash:
9   hosts: ["<direccion>:<puerto>"]
10
11 processors:
12   - add_host_metadata:
13     when.not.contains.tags: forwarded
14   - add_cloud_metadata: ~
15   - add_docker_metadata: ~
16   - add_kubernetes_metadata: ~
```

El resto del fichero puede estar comentado.

Para finalizar activaremos el módulo system de Filebeat y indicaremos los logs del sistema que queremos enviar:

```
1 #Activar el modulo system
2 filebeat modules enable system
3 #Comprobar los modulos activados
4 filebeat modules list
```

Ya activado debemos de modificar el fichero correspondiente que se encontrará en `/etc/filebeat/modules.d/system.yml`

```
1 # Module: system Docs: https://www.elastic.co/guide/en/beats/filebeat/master/
   filebeat-module-system.html
2
3 - module: system
4   # Syslog
5   syslog:
6     enabled: true
7     var.paths: ["/var/log/syslog"]
8
9   auth:
10    enabled: true
11    var.paths: ["/var/log/auth.log"]
```



## Anexo B: Manual de uso.

En este documento se explicará el uso básico del panel de OpenSearch, es decir, manejo de índices, creación de patrones de índices, creación de dashboard y creación de visualizaciones.

### Manejo de índices

Si nos dirigimos al apartado **Index Management** > **Índices** podremos ver todos los índices, ver el estado de salud y realizar ciertas operaciones con ellos:

- Crear un nuevo índice: Puedes crear un nuevo índice proporcionando un nombre y una configuración específica. Esto te permite organizar y almacenar tus datos de manera adecuada.
- Listar los índices existentes: OpenSearch Dashboards te muestra una lista de todos los índices disponibles en tu clúster. Puedes ver información como el nombre, tamaño y número de documentos en cada índice.
- Ver estadísticas del índice: Puedes acceder a estadísticas detalladas de un índice en particular, como el número de documentos indexados, tamaño del índice, estadísticas de búsqueda y más.
- Actualizar la configuración del índice: Puedes modificar la configuración de un índice existente, como los análisis de texto, los campos indexados o las configuraciones de retención de datos.
- Eliminar un índice: Si ya no necesitas un índice, puedes eliminarlo para liberar espacio y recursos en tu clúster. Sin embargo, debes tener cuidado al hacerlo, ya que esto resultará en la pérdida permanente de los datos almacenados en ese índice.
- Administrar políticas de ciclo de vida (Lifecycle Policies): Puedes configurar políticas de ciclo de vida para tus índices. Estas políticas definen acciones automáticas, como la eliminación o el archivado de datos después de cierto tiempo o tamaño. Esto te permite administrar el crecimiento y el mantenimiento de los índices de manera eficiente.

Index	Health	Managed by policy	Status	Total size	Size of primaries	Total documents	Deleted documents	Primaries	Replicas
system-logs-2023.06.18	Green	No	Open	39.1mb	39.1mb	119816	0	1	1
security-auditlog-2023.06.18	Green	No	Open	491.3kb	491.3kb	165	0	1	1
security-auditlog-2023.06.13	Green	No	Open	780.8kb	780.8kb	766	0	1	1
.opensearch-observability	Green	No	Open	6kb	6kb	1	0	1	1
.opensearch-notifications-config	Green	No	Open	208b	208b	0	0	1	1
.opendistro_security	Green	No	Open	71.7kb	71.7kb	10	0	1	1
.opendistro-reports-instances	Green	No	Open	6.1kb	6.1kb	1	0	1	1

Figura 17: Anexo B: Manejo de índices

### Patrones de índices

Para crear un “Index Pattern” en OpenSearch Dashboards debmos de seguir estos pasos:

- Nos dirigimos a la sección “Stack Management” en el menú lateral.
- Hacer clic en el botón “Create index pattern” (Crear patrón de índice).



- En el campo “Index pattern” (Patrón de índice), ingresa el nombre o patrón que quieres utilizar para identificar múltiples índices. El patrón puede contener comodines como asteriscos para que coincida con varios índices con nombres similares. Por ejemplo, si existen índices con nombres como "logs-2023-01-01", "logs-2023-01-02", etc., se puede ingresar "logs-¿como patrón.
- En el campo “Time Filter field name” (Nombre del campo de filtro de tiempo), selecciona el campo de fecha o timestamp que se utilizará para filtrar y visualizar datos en un rango de tiempo específico. Este campo debe existir en los índices que coinciden con el patrón.
- Haz clic en el botón “Create index pattern” (Crear patrón de índice) para crear el patrón de índice.

El “Index Pattern”(Patrón de índice) se utiliza en OpenSearch Dashboards para definir cómo se accede y visualiza la información de los índices relacionados. Proporciona una forma de agrupar y consultar datos de múltiples índices de manera eficiente.

Al crear un “Index Pattern”, OpenSearch Dashboards va a realizar las siguientes tareas:

- Identifica y registra los índices que coinciden con el patrón proporcionado.
- Extrae información sobre los campos y tipos de datos disponibles en los índices.
- Configura la visualización de datos, lo que permite utilizar herramientas como gráficos, tablas y visualizaciones personalizadas para explorar y analizar los datos contenidos en los índices.
- Habilita la funcionalidad de búsqueda y filtrado de datos utilizando el campo de filtro de tiempo especificado.

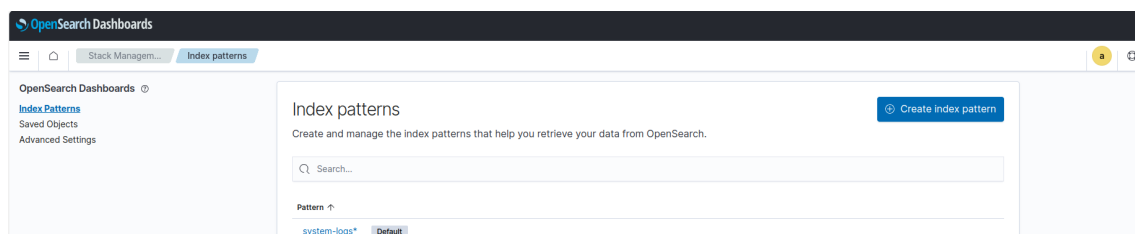


Figura 18: Anexo B: Patrones de índices

## Discover

El apartado “Discover” en OpenSearch Dashboards proporciona una interfaz para explorar y buscar datos en los índices. Accederemos desde el menú lateral pulsado en la pestaña “Discover”. El uso de esta interfaz es la siguiente:

- Selección del índice: En la parte superior de la página, existe un desplegable donde puedes seleccionar el índice o el patrón de índice que deseas explorar. Puedes elegir entre los patrones de índice que hayas creado previamente o seleccionar un índice específico
- Vista de documentos: En la parte central de la página, verás una tabla que muestra los documentos de los índices seleccionados. Cada fila representa un documento individual y cada columna corresponde a un campo dentro del documento.
- Búsqueda y filtrado: En la parte superior de la página, encontraremos una barra de búsqueda que permite buscar datos específicos en los documentos. Se puede ingresar consultas de búsqueda para filtrar los resultados en función de los valores de los campos. También podemos aplicar filtros adicionales para refinar aún más los resultados.





- Guardado de búsquedas: Si se desea conservar una búsqueda específica para referencia futura, nos permite guardarla haciendo clic en el botón “Save” (Guardar) en la barra de herramientas superior. Esto te permite acceder rápidamente a la búsqueda guardada en cualquier momento.

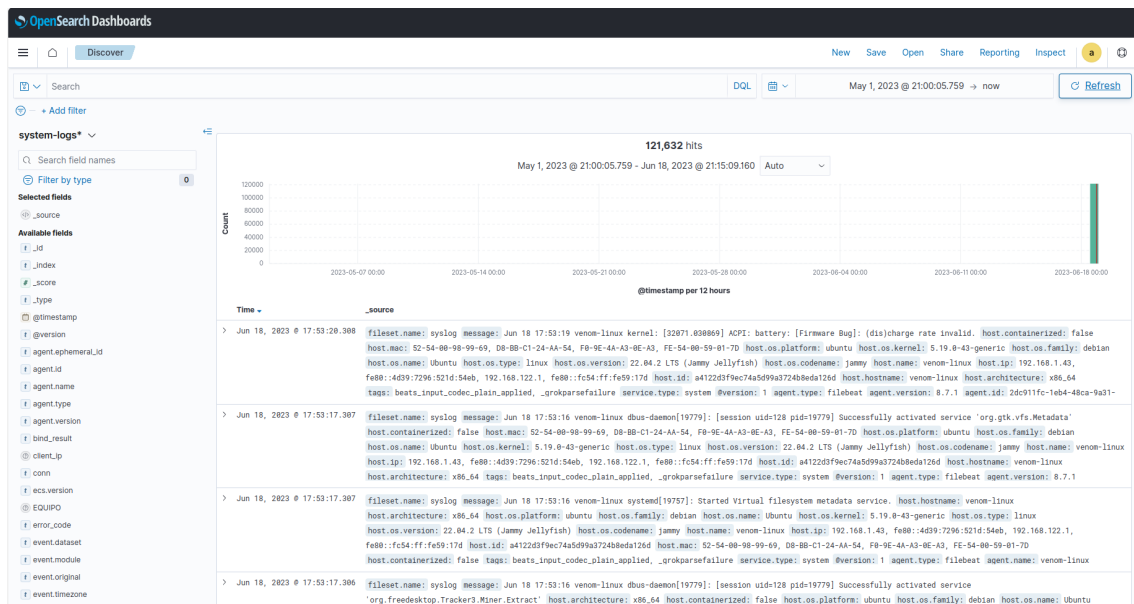


Figura 19: Anexo B: Discover

## Dashboard

OpenSearch Dashboards es una sección que nos permite crear paneles de control personalizados para visualizar y analizar los datos de manera visual y práctica. El funcionamiento es el siguiente:

- Acceso a Dashboard: En el menú lateral de OpenSearch Dashboards, haz clic en la pestaña “Dashboard” para acceder a esta herramienta.
- Creación de un nuevo panel: Puedes comenzar creando un nuevo panel haciendo clic en el botón “Create dashboard” (Crear panel) o seleccionando uno existente si ya existe uno.
- Diseño del panel: Una vez en el panel, se puede diseñar la apariencia y contenido. Podemos agregar elementos como gráficos, visualizaciones, tablas, métricas, filtros y otros componentes interactivos. Estos elementos se conocen como "visualizaciones" permiten representar tus datos de diversas formas.
- Configuración de las visualizaciones: Cada visualización en el panel se puede configurar para mostrar datos específicos de los índices. Podemos elegir el índice, definir consultas de búsqueda y aplicar filtros para limitar los datos que se muestran en cada visualización.
- Organización del panel: Es posible organizar las visualizaciones en el panel según las necesidades. Se puede mover y redimensionar las visualizaciones, crear múltiples páginas o pestañas dentro del panel y establecer filtros interactivos para que los datos se actualicen automáticamente.
- Guardado y compartición del panel: Una vez que el panel esté diseñado es posible guardarlo para acceder a él posteriormente. También se puede compartir con otros usuarios para que puedan ver el mismo panel de control y beneficiarse de la visualización de datos.
- Personalización y configuración adicional: OpenSearch Dashboards ofrece varias opciones de personalización y configuración para los paneles. Podemos ajustar la apariencia visual, configurar paneles en modo de pantalla completa, establecer opciones de actualización automática y más.

## CFGS 2º ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS Y REDES

Centralación y análisis de logs mediante Opensearch

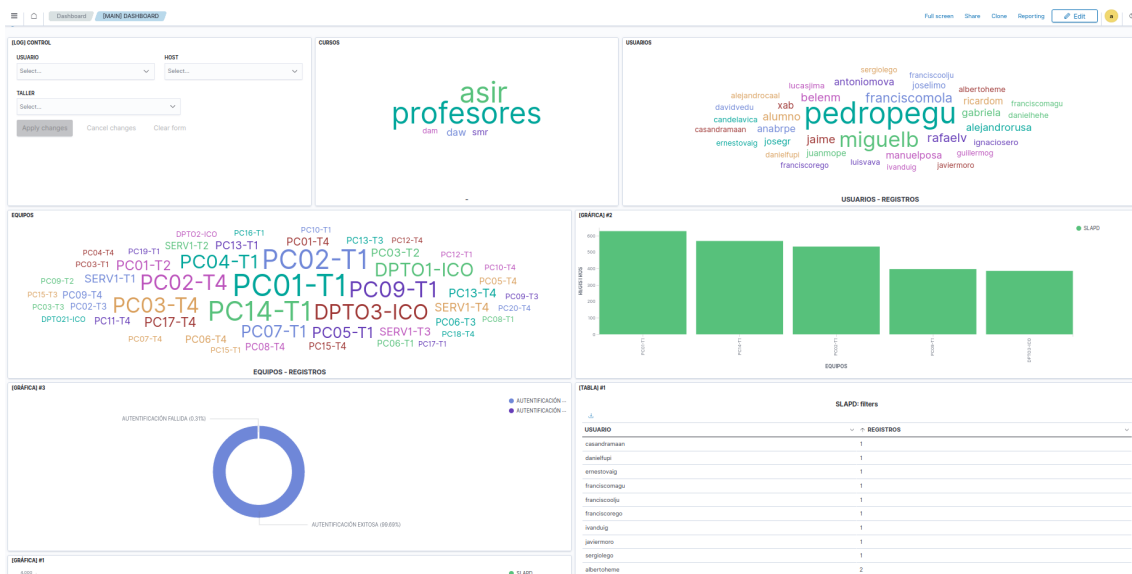


Figura 20: Anexo B: Dashboard

## Visualizations

El apartado “Visualizations” en OpenSearch Dashboards es una sección que permite crear y personalizar visualizaciones de datos para mostrar información de manera gráfica y comprensible.

- **Acceso a Visualizations:** En el menú lateral de OpenSearch Dashboards, hacer clic en la pestaña “Visualizations” para acceder a esta herramienta.
- **Creación de una nueva visualización:** Creamos una nueva visualización haciendo clic en el botón “Create visualization” (Crear visualización) o seleccionando una visualización existente si ya existe alguna creada.
- **Selección del tipo de visualización:** OpenSearch Dashboards ofrece una variedad de tipos de visualizaciones, como gráficos de barras, gráficos de líneas, tablas, gráficos circulares y más. Podemos seleccionar el tipo de visualización que mejor se adapte a tus datos y objetivos.
- **Configuración de la fuente de datos:** Para cada visualización, se debe configurar la fuente de datos, es decir, el índice o patrón de índice que contiene los datos que deseas visualizar. Puedes definir consultas de búsqueda y filtros para limitar los datos que se mostrarán en la visualización.
- **Configuración de los campos y métricas:** Puedes seleccionar los campos específicos que se desea visualizar y definir las métricas o agregaciones que se aplicarán a esos campos. Por ejemplo, puedes mostrar la suma, el promedio o el recuento de los valores de un campo en particular.
- **Personalización de la apariencia visual:** Es posible personalizar la apariencia visual de la visualización, como el color, el tamaño, los ejes, las etiquetas y otros elementos visuales. Esto te permite adaptar la visualización a tus preferencias y requisitos específicos.
- **Guardado de la visualización:** Una vez que se haya configurado y personalizado la visualización, podemos guardarla para acceder a ella posteriormente y reutilizarla en diferentes paneles o dashboards.
- **Compartición de la visualización:** OpenSearch Dashboards permite compartir visualizaciones con otros usuarios, lo que les permite ver y utilizar las mismas visualizaciones en sus propios paneles y dashboards.

# CFGS 2º ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS Y REDES

Centralación y análisis de logs mediante Opensearch

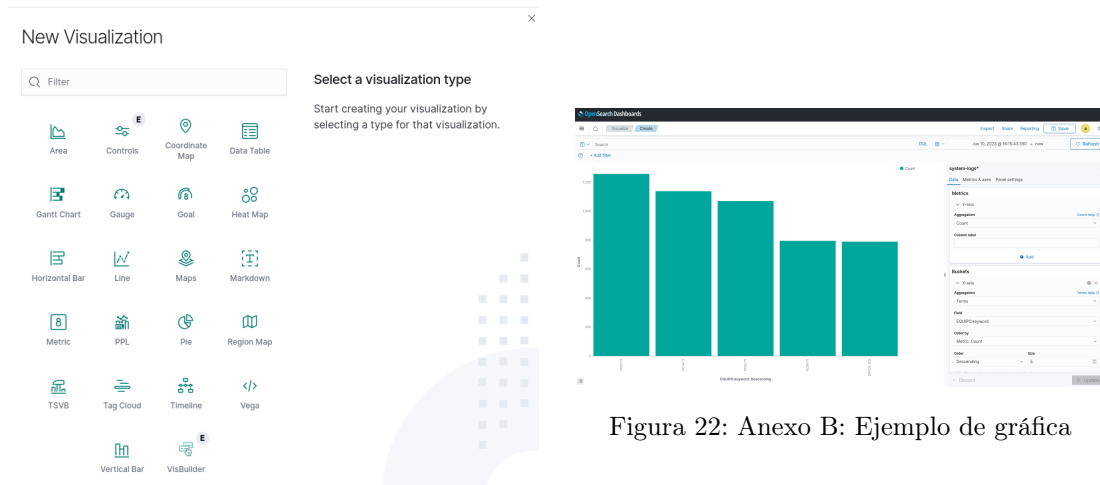


Figura 22: Anexo B: Ejemplo de gráfica

Figura 21: Anexo B: Crear una visualización