



OPENSEARCH

SMDB no relacional.

Sánchez Valenzuela Natalia

Fei Fei Wu Zhang

Sandoval López Daniela

ÍNDICE

- | | | |
|----------------------------|------------------------------------|------------------------|
| 1 Introducción | 6 Métricas | 11 Aplicaciones |
| 2 Historia y Origen | 7 Funciones | 12 Conclusión |
| 3 Contribuyentes | 8 Manejo DDL/DML | 13 Referencias |
| 4 Objetivos | 9 Compatibilidad | |
| 5 Características | 10 Lenguaje de programación | |

INTRODUCCIÓN

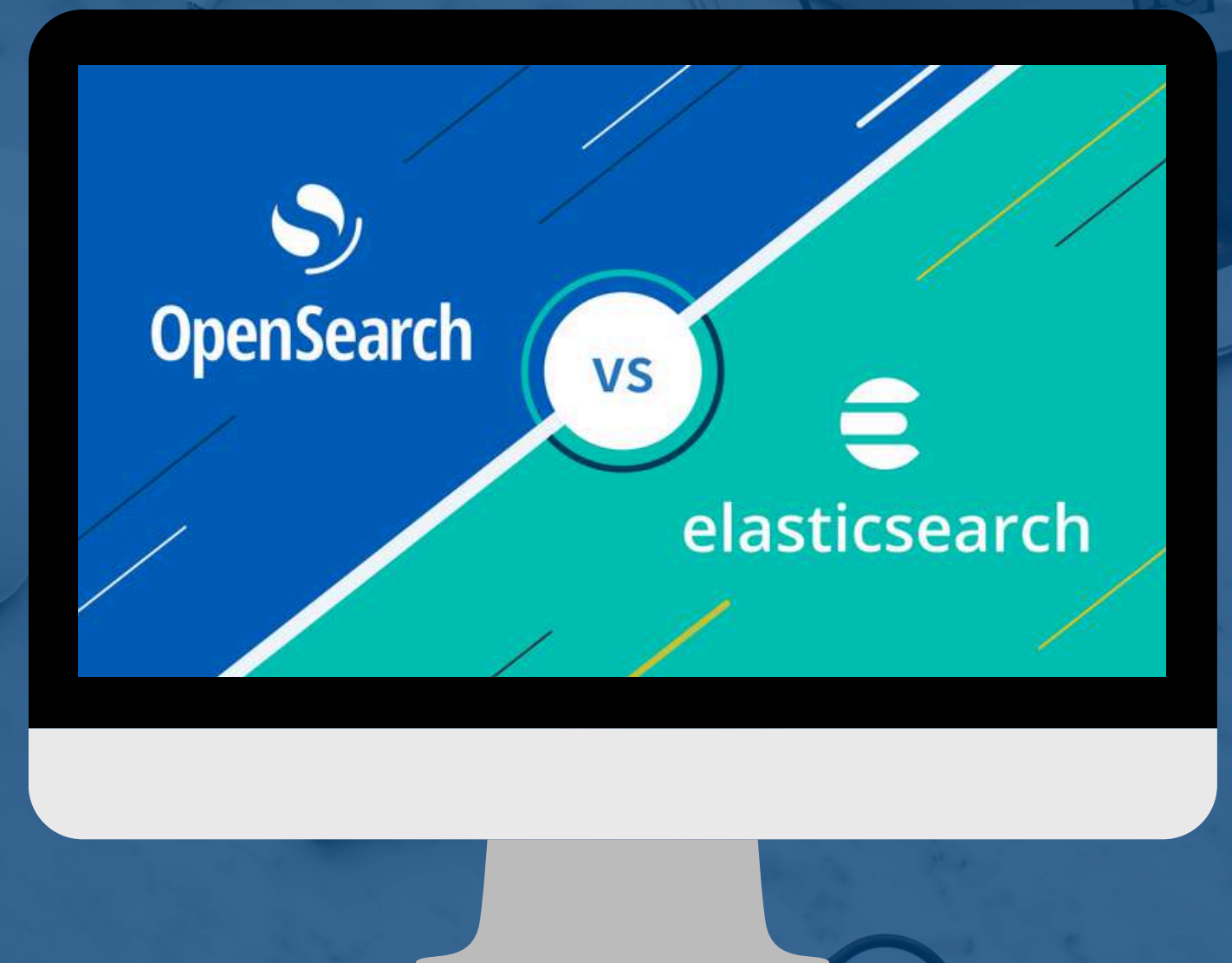
OpenSearch es un **motor de búsqueda** y análisis distribuido de **código abierto**, desarrollado a partir de **Elasticsearch**. Proporciona una solución robusta para la **ingesta, indexación, búsqueda y análisis de grandes volúmenes de datos**. Su propósito principal es permitir búsquedas rápidas y eficientes, así como la visualización y el análisis de datos en tiempo real, transformando la manera en que trabajamos con grandes volúmenes de información.

HISTORIA Y ORIGEN

Fue en **enero de 2021** cuando Amazon Web Services (AWS) anunció el proyecto como una **bifurcación** (fork) de los populares **Elasticsearch y Kibana**.

En **julio de 2021**, se lanzó oficialmente **OpenSearch 1.0**. Este lanzamiento fue un hito importante, ya que el proyecto se publicó bajo la Licencia Apache, Versión 2.0 (ALv2), garantizando que todo su **código fuera 100% abierto**.

En **noviembre de 2022**, la **versión 2.4** se lanzó con mejoras significativas, incluyendo **soporte para Windows**, mayor resiliencia del clúster y **nuevas funcionalidades** de búsqueda.




CONTRIBUYENTES

Aunque Amazon Web Services (AWS) fue el principal impulsor de su lanzamiento, OpenSearch es un **proyecto impulsado por la comunidad de código abierto**.

Cuenta con la participación activa de **miles de desarrolladores** y organizaciones que contribuyen a su crecimiento.



OBJETIVOS INICIALES



Garantizar la Apertura: Mantener una solución de **búsqueda y análisis que fuera 100% de código abierto** bajo la Licencia Apache 2.0 (ALv2).

Fomentar la Colaboración: Permitir que la comunidad pudiera **innovar y desarrollar nuevas funciones sin restricciones**.

Asegurar la Flexibilidad: Ofrecer a los usuarios la **libertad de usar, modificar** y distribuir la plataforma como mejor les parezca.

CARACTERÍSTICAS ÚNICAS



Seguridad y Detección de Anomalías

OpenSearch cuenta con seguridad integrada que incluye **autenticación y cifrado**. Además, tiene un sistema de **detección de anomalías** que utiliza algoritmos para identificar patrones inusuales en los datos, algo vital para el monitoreo y la protección.



Visualización y Análisis Integrados

Gracias a OpenSearch Dashboards, la plataforma ofrece una **visualización integrada** con paneles de control intuitivos. Esto te permite **explorar y analizar los datos directamente**, sin necesidad de usar herramientas externas.



Ingesta y Preparación Eficiente de Datos

Mediante el complemento Data Prepper, OpenSearch **facilita una ingesta robusta de datos** y permite transformarlos y enriquecerlos directamente desde el servidor.

MÉTRICAS DE RENDIMIENTO Y EFICIENCIA

Velocidad de Indexación:

Se refiere a la **rapidez** con la que OpenSearch puede **agregar nuevos datos**. Gracias a su arquitectura distribuida, **puede procesar millones de documentos por segundo**, lo que es fundamental para aplicaciones que requieren análisis en tiempo real, como el monitoreo de logs.



Velocidad de Búsqueda:

Mide el tiempo que tarda el sistema en devolver los resultados de una consulta. OpenSearch **está optimizado para proporcionar respuestas casi instantáneas**, incluso en conjuntos de datos de petabytes.

FUNCIONES

1

ALMACENAMIENTO, RECUPERACIÓN Y ACTUALIZACIÓN DE DATOS



Tecnologías y Mecanismos: OpenSearch organiza los datos en **índices que se dividen en shards** (fragmentos). Estos se distribuyen en un clúster de nodos.

Procesos Específicos: El sistema utiliza una arquitectura distribuida que **facilita la búsqueda y el almacenamiento** de grandes volúmenes de datos de manera eficiente.

Impacto en el Usuario: **El usuario puede insertar, buscar y actualizar** datos en tiempo real, ya que el sistema gestiona la complejidad de la distribución y el balanceo de carga.

FUNCIONES

2

SERVICIOS DE AUTORIZACIÓN



Tecnologías y Mecanismos: OpenSearch utiliza un sistema de control de acceso basado en roles (RBAC). **Los roles definen los permisos de un usuario** para acceder a índices o documentos.

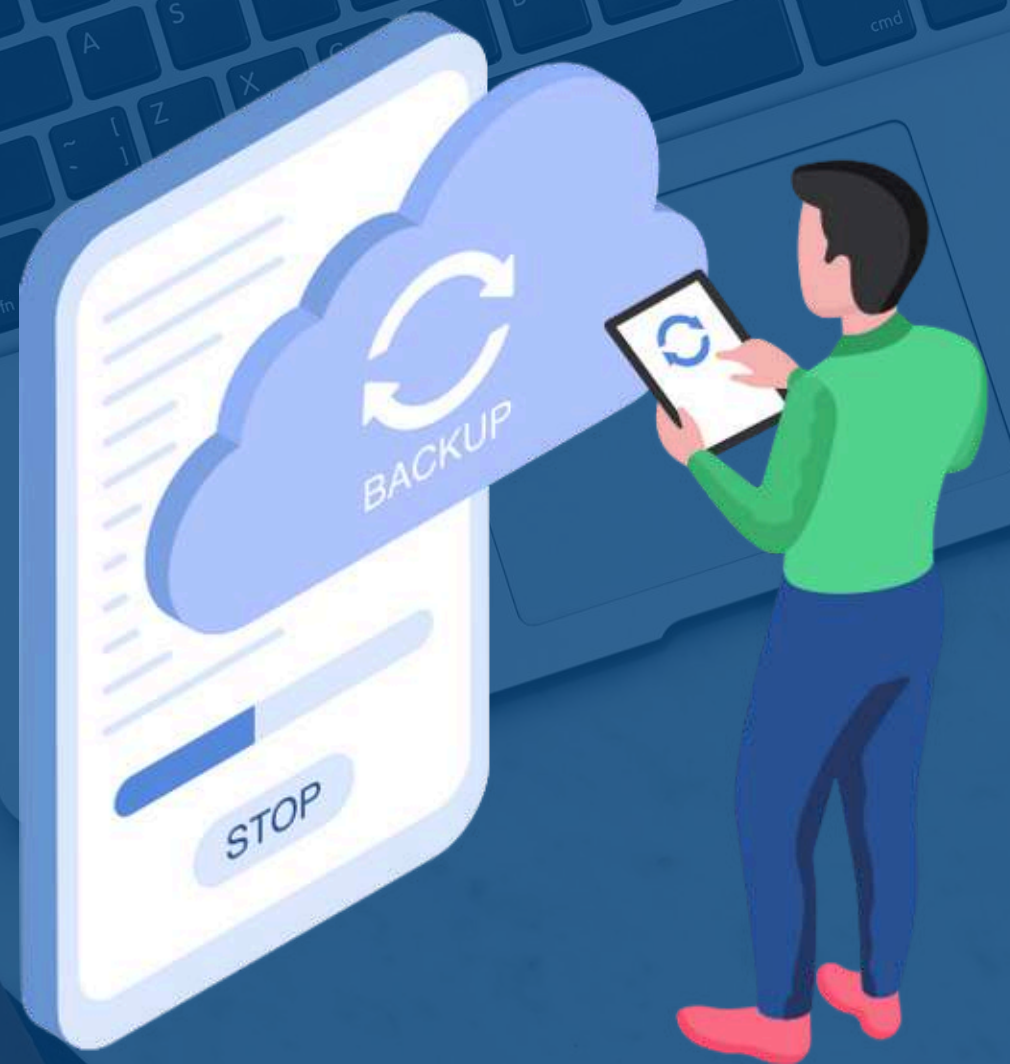
Procesos Específicos: **El sistema verifica los permisos** antes de permitir cualquier operación de lectura o escritura, lo que garantiza la seguridad de los datos.

Impacto en el Usuario: Permite a los administradores **controlar quién puede ver y modificar la información**, lo que es crucial para la seguridad y privacidad.

FUNCIONES

3

SERVICIOS DE RECUPERACIÓN



Tecnologías y Mecanismos: OpenSearch **permite crear snapshots** (instantáneas) de los índices, que son **copias de seguridad** de los datos.

Procesos Específicos: Estos **snapshots se almacenan en repositorios remotos** y se pueden restaurar para recuperar el clúster a un estado anterior.

Impacto en el Usuario: Proporciona una manera confiable de hacer copias de seguridad y de recuperar la información en caso de fallos, lo que **garantiza la durabilidad de los datos**.

FUNCIONES

4

SERVICIOS DE CONTROL DE CONCURRENCIA



Tecnologías y Mecanismos: El sistema utiliza un **modelo de concurrencia optimista** que usa versiones de documentos en lugar de bloqueos.

Procesos Específicos: Si varios usuarios intentan actualizar un mismo documento, solo **la primera actualización es aceptada**. Las siguientes son rechazadas para evitar conflictos.

Impacto en el Usuario: Permite que múltiples usuarios realicen **operaciones simultáneas sin bloqueos**, lo que mejora significativamente el **rendimiento**.

FUNCIONES

5

SERVICIOS DE INTEGRIDAD



Tecnologías y Mecanismos: La integridad se maneja a través del **mapping**, un proceso que **define la estructura y el tipo de dato de cada campo**.

Procesos Específicos: Durante la indexación, el sistema valida que **los nuevos datos coincidan con el mapping**. Si no es así, la operación es rechazada.

Impacto en el Usuario: Asegura que **la información almacenada** en la base de datos **sea consistente y evite datos corruptos**.

MANEJO DE DDL Y DML

En ciertas ediciones de OpenSearch, **no** se admiten comandos **DDL** (como **CREATE**, **ALTER**, **DROP**) ni **DML** (como **INSERT**, **UPDATE**, **DELETE**).

La **creación** de índices y mappings en OpenSearch se hace mediante la **API REST**, no con comandos DDL tradicionales.

Para **añadir** o **actualizar** un documento se usa el **PUT**.

Para **eliminar** un documento se usa **DELETE**.

OpenSearch ofrece varios lenguajes para buscar y analizar:

- **Query DSL**: lenguaje en **JSON**, muy **flexible** y **potente**.
- **DQL (Dashboards Query Language)**: **consultas** simples desde la **interfaz** de OpenSearch Dashboards.
- **SQL plugin**: permite usar **SELECT**, **WHERE**, **GROUP BY**, pero solo para consultas de lectura, no para modificar datos.
- **PPL (Piped Processing Language)**: estilo de **comandos** en tubería, útil para **análisis de logs y métricas**.



COMPATIBILIDAD DEL SISTEMA OPERATIVO

Puede implementarse en una variedad de sistemas operativos. Para funcionar, requiere que el entorno tenga Java instalado.



Linux: Es el entorno de producción **más recomendado y optimizado**. La distribución oficial es ideal para sistemas como Ubuntu, CentOS y Red Hat, donde se espera un **alto rendimiento**.



Windows: Funciona bien para **entornos de desarrollo y pruebas**. No se considera la opción ideal para **cargas de trabajo de producción de alta demanda**.



macOS: Es un excelente entorno para que los desarrolladores realicen **pruebas y experimenten** con la plataforma. Al igual que Windows, no es el más adecuado para **despliegues de producción a gran escala**.

SOPORTE DE LENGUAJE DE PROGRAMACIÓN

Llenguajes_y_descripciones.txt

1. Java

A través del Cliente de Alto Nivel de OpenSearch. Este cliente se encarga de **serializar y deserializar** las peticiones y respuestas en objetos Java, maneja el **descubrimiento de nodos y el balanceo de carga**, y es compatible con las versiones principales de OpenSearch

2. Python

A través del cliente oficial de Python, que proporciona una abstracción sobre las solicitudes HTTP a la API REST. Los métodos del cliente **mapean directamente con los endpoints** de la API de OpenSearch.

3. JavaScript

Similar a Python, el cliente de Node.js envía solicitudes HTTP a la API de OpenSearch y procesa las respuestas JSON, todo envuelto en una API JavaScript.

4. Go

Ayuda a **ingerir o consultar grandes volúmenes de datos de forma eficiente** a través del cliente, ya que proporciona una API tipada para construir peticiones y parsear respuestas.

5. .NET (C#)

Utilizando el cliente oficial de .NET, que sigue la misma filosofía que los otros clientes oficiales: **una API fuertemente tipada que abstrae las llamadas HTTP**.

SOPORTE DE LENGUAJE DE PROGRAMACIÓN

Bibliotecas_o_marcos.txt

1. Java

OpenSearch Java Client: El cliente oficial de alto nivel. Recomendada para nuevas aplicaciones.

Spring Data OpenSearch: Proporciona una abstracción de repositorio familiar para los desarrolladores de Spring, permite interactuar con OpenSearch usando interfaces de repositorio y objetos de dominio (*POJOs*) con anotaciones, minimizando el código.

2. Python

opensearch-py: El cliente oficial de Python para OpenSearch. Es la opción principal y más segura.

Elasticsearch DSL: Facilita la escritura de consultas de manera más programática y menos verbosa que un JSON crudo.

3. JavaScript

opensearch-js: El cliente oficial de JavaScript/Node.js para OpenSearch.

Elasticsearch JS (v7.13+)

4. Go

opensearch-go: El cliente oficial de Go para OpenSearch.

5. .NET (C#)

OpenSearch .NET Client: El cliente oficial para .NET.

APLICACIONES DEL MUNDO REAL



Análisis de Registros (Logs)

Empresas como **Netflix y Uber** lo utilizan para **procesar y analizar millones de logs** de sus servidores en tiempo real.

Esto les permite **detectar errores** y problemas de rendimiento de manera instantánea, **garantizando la estabilidad** de sus servicios.



Búsqueda en Sitios Web

Grandes plataformas de comercio electrónico, como **Amazon**, utilizan tecnologías similares para **potenciar sus funciones de búsqueda**.

OpenSearch permite a los usuarios encontrar productos de manera rápida y relevante en catálogos masivos, lo que **mejora la experiencia de compra**.



Monitoreo de Aplicaciones (APM)

En la industria, se usa para **monitorear el rendimiento de las aplicaciones**.

Al recopilar métricas y rastrear transacciones, OpenSearch ayuda a los equipos a **identificar y resolver problemas antes de que afecten a los usuarios**.

CONCLUSIÓN

En equipo, y basándonos en la investigación que realizamos, concluimos que OpenSearch es una **herramienta** sumamente **útil y poderosa**. Su modelo de código abierto, su gran escalabilidad y sus características de seguridad y visualización la convierten en una **solución ideal para el análisis de datos**.

En definitiva, OpenSearch no es solo un motor de búsqueda, es una solución integral para la toma de decisiones que las empresas utilizan activamente para **convertir grandes volúmenes de datos en información valiosa y accesible**.

REFERENCIAS

[https://aws-amazon-com.translate.goog/what-is/opensearch/?
_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc](https://aws.amazon-com.translate.goog/what-is/opensearch/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

[https://opensearch-org.translate.goog/about/?
_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc](https://opensearch-org.translate.goog/about/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

<https://aws.amazon.com/es/opensearch-service/features>

[https://aws-amazon-com.translate.goog/blogs/big-
data/improve-your-amazon-opensearch-service-performance-
with-opensearch-optimized-instances/?
_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc](https://aws-amazon-com.translate.goog/blogs/big-data/improve-your-amazon-opensearch-service-performance-with-opensearch-optimized-instances/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)



GRACIAS