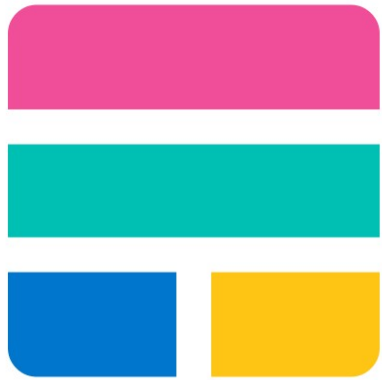


Arquitectura Elastic Stack.



elastic stack

Índice de contenidos

1. Nodos
2. Índices y documentos
3. Shards y réplicas
4. Hardware recomendado

1. Nodos (I)

- Son cualquier instancia / servidor de *Elasticsearch*.
- Varios nodos conectados forman un *cluster*.
- Los nodos son los encargados del tráfico HTTP en la **capa de aplicación** de la pila **ISO/OSI** y también se encarga de la **capa de transporte** (conexiones TCP/UDP/SPX) para la comunicación interna de los nodos del *cluster*.
- Reparten funcionalidades. Tipos de nodos:
 - Master Node
 - Data Node
 - Ingest Node
 - Coordinating only Nodes

1. Nodos (II). Master Node

Responsable de acciones como:

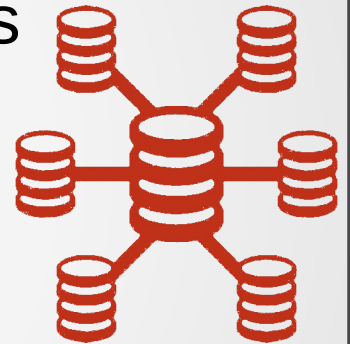
- Controla que todo funcione en el *cluster*.
- Crear / eliminar índices.
- Controlar qué nodos forman parte del *cluster*.
- Decidir en que nodo alojar los datos.

Nota: Es necesario que el Master Node sea un equipo dedicado, lo más estable posible, que tenga alta disponibilidad y no sufra caídas.

1. Nodos (III). Data Nodes

Encargados de **almacenar** la información y datos

- Operaciones CRUD sobre los datos almacenados
- Realizar búsquedas sobre los datos almacenados
- Realizar **agregaciones** sobre el dominio



Nota: Estos equipos tienen un gran consumo de recursos como CPU, RAM y operaciones I/O.

Se recomienda tener *Data Nodes* dedicados y que haya un **crecimiento horizontal**.

1. Nodos (IV). Ingest Node

Ejecutan tareas de **preprocesamiento** antes de indexar la información.

Tratan la información obtenida para formatearla, separarla, en definitiva ajustarla para ser almacenada con un sentido lógico y coherente.

Cuando tenemos que procesar, formatear o enriquecer cientos o miles de datos en un segundo, vamos a necesitar alta capacidad de procesamiento.

Nota: Si se trabaja con altas tasas de datos, se recomienda que el *Ingest Node* sea un equipo dedicado.

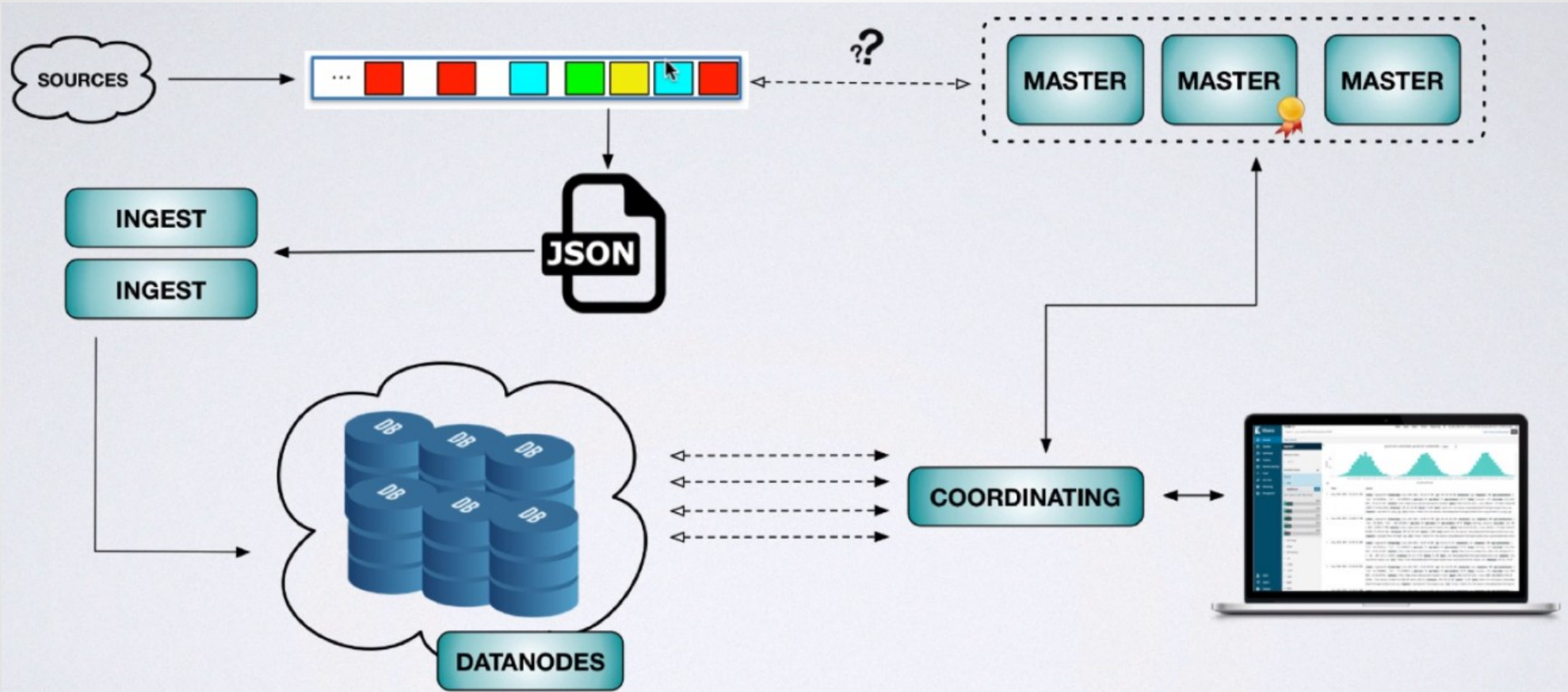
1. Nodos (V). Coordinating Node

Solo serán necesarios si tenemos que gestionar gran cantidad de nodos, distintos flujos e índices, procesar grandes cantidades de datos y con un nodo máster sobrecargado.

Son **balanceadores de carga** que ayudan a descargar al Master Node. Tareas:

- Encaminar búsquedas
- Distribuir indexados

1. Nodos dispuestos en *cluster*



2. Índices y Documentos (I). Documentos

- Documento/Evento:
 - Unidad básica de información que puede ser indexada.
 - Formato JSON: <clave:valor>
- Compuesto por: **Datos** y **Metadatos**.
 - *_index*: Donde los documentos/eventos residen.
 - *_type*: Clase de objeto que el documento representa.
 - *_id*: Identificador único del documento/evento.

```
{  
  "Title": "Sweet Lady",  
  "Duration": "04:01",  
  "Writer": "Brian May"  
},
```

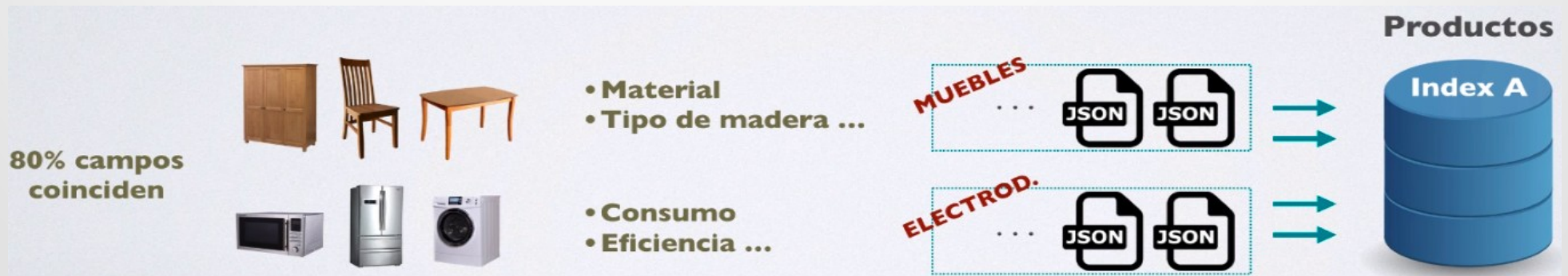
2. Índices y Documentos (II). Índices

- Índice:
 - Colección de documentos con características similares (catálogo de productos, clientes, pedidos, facturas...)
 - Análogo a una tabla en bases de datos relacionales pero sin datos estructurados.



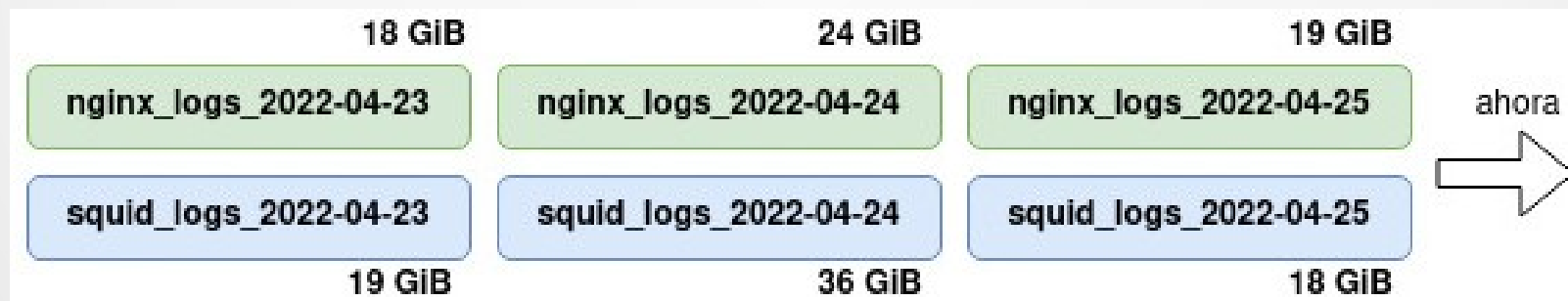
2. Índices y Documentos (III). Types

- *Types*:
 - Cada *Type* representa una clase de documentos similares (relaciones entre muebles)
 - Acompañado de un *mapping* (esquema) que describe:
 - Campos que lo componen
 - *Datatype* de cada uno (*string*, *integer*, *date*...)



2. Índices y Documentos (II). Índices Logging

- *Elasticsearch* es ampliamente usado para Logging
- Organizados en tiempo: mejora en rendimiento.
- Miles de dispositivos → GiB / TiB Logs → Big Data



3. Shards y réplicas (I). Problemática Big Data

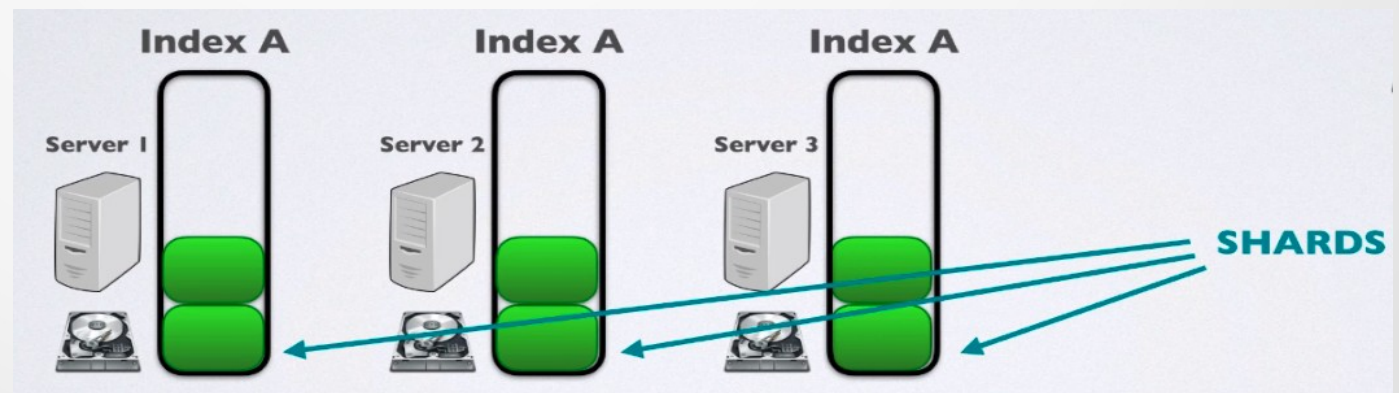
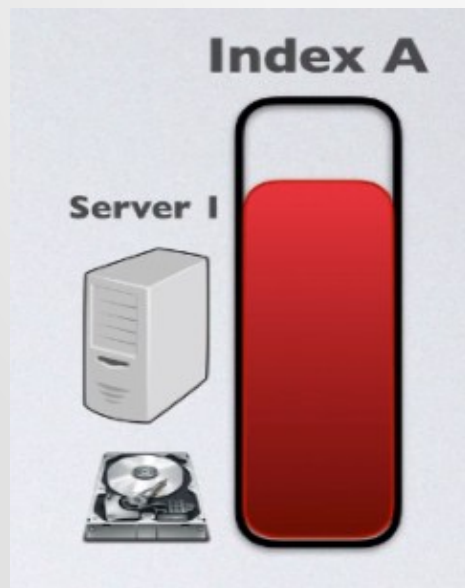
Puntos a resolver con la gestión de información:

- Gran cantidad de información = límite de HW → Upgrade!
 - +Almacenamiento → +CPU y +RAM
- Sobrecarga de un único SGBD = límite de SW.
- Caída del sistema o del SO:
 - Pérdida del servicio
 - Posible pérdida de datos
 - Desconfianza de los usuarios → Pérdida de clientes

3. Shards y réplicas (II) Solución BigData

Subdividir el índice en partes denominadas **Shards**

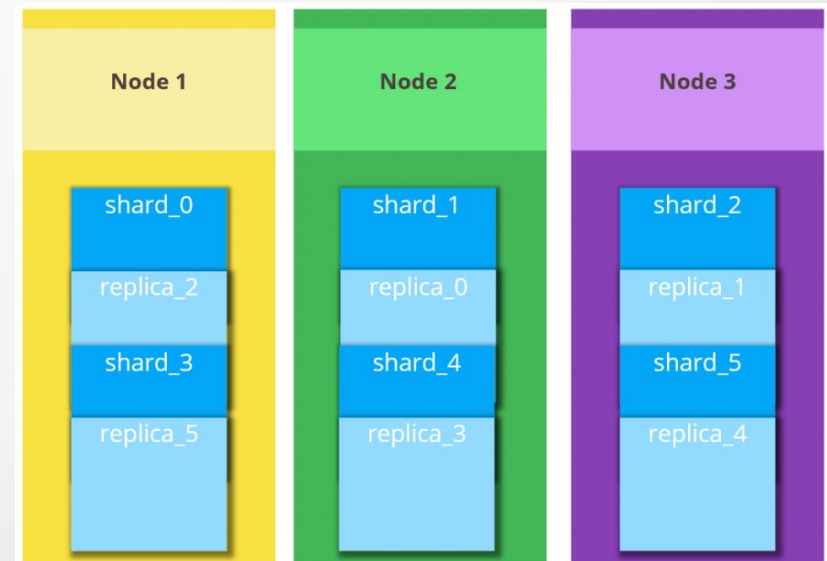
- Las operaciones son distribuidas y paralelas
- Se reparte la carga computacional
- Todavía existen problemas debido a que si se cae un *Shard*, perderíamos datos. No tenemos redundancia.



3. Shards y réplicas (III). Réplicas

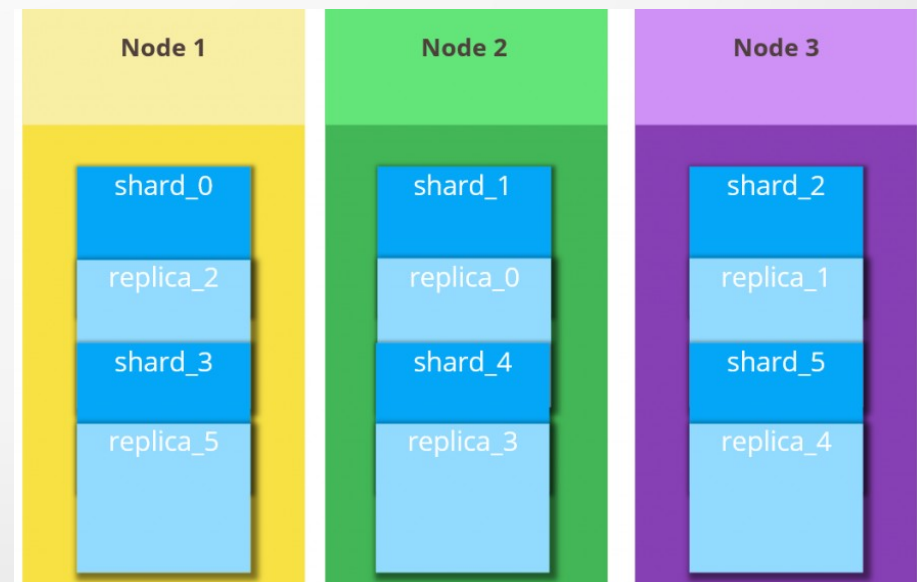
Réplica = Copia de un *Shard*

- Alta disponibilidad y tolerancia a fallos (*Shards* y nodos)
- Índices compuestos por: *Shards* primarios y sus correspondientes réplicas
- La réplica no estará en el mismo nodo
- Permiten búsquedas en paralelo



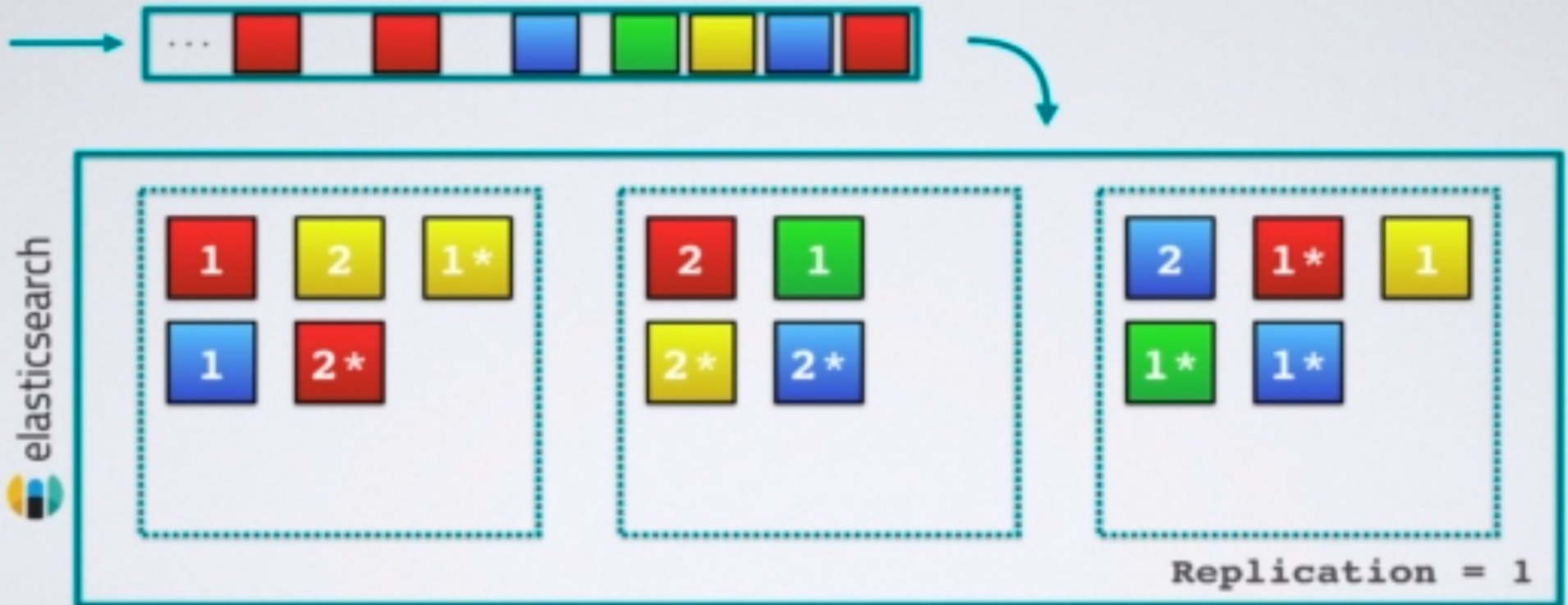
3. Shards y réplicas (IV). BD distribuidas

- La información se almacena en distintos puntos de red.
- Transparente para el usuario final.
- Independencia respecto al SO y al HW.
- Procesamiento distribuido de consultas.
- Réplicas = alta disponibilidad (Sistemas en producción).



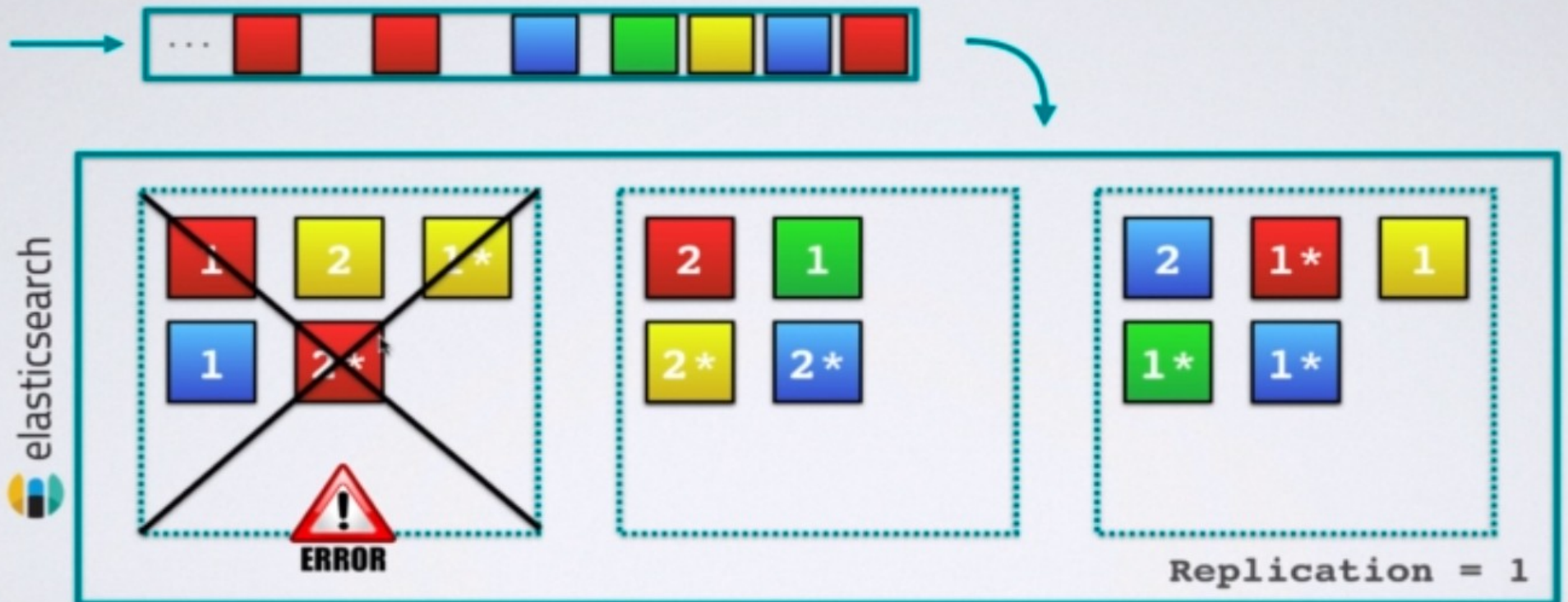
3. Shards y réplicas (V). Ejemplo (I)

INDEXADO DE INFORMACIÓN



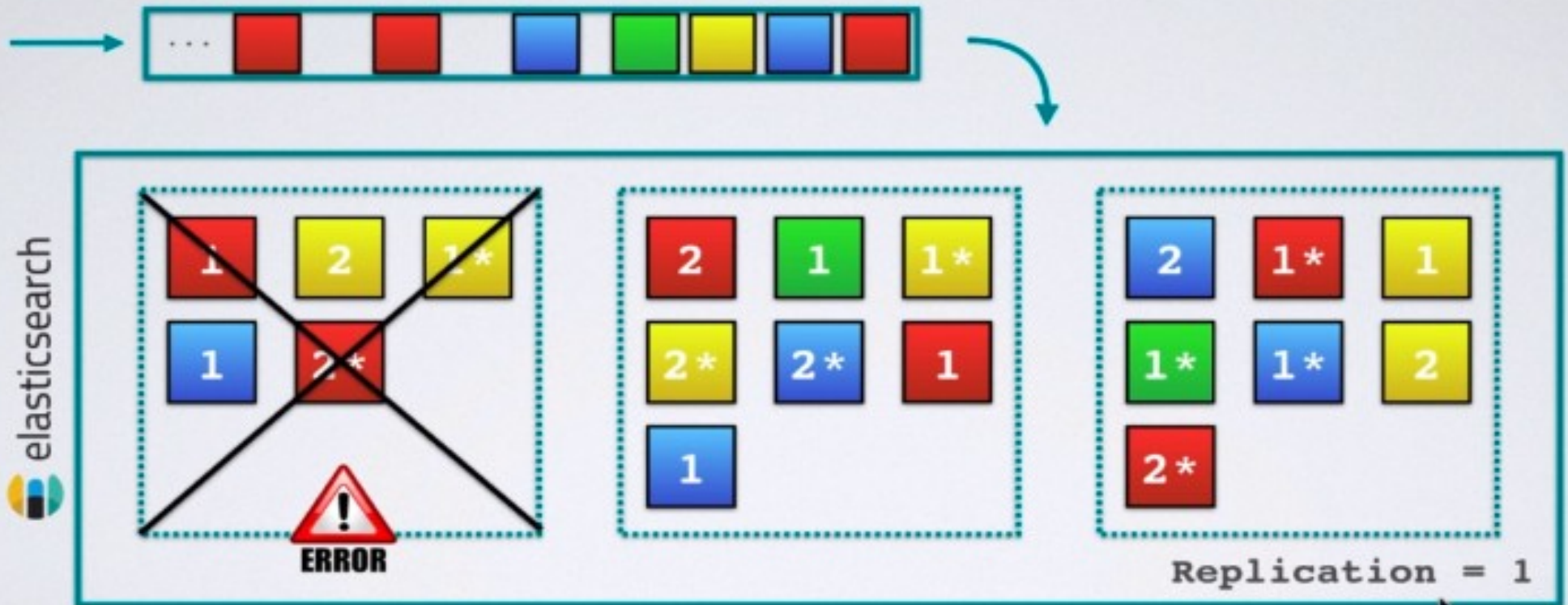
3. Shards y réplicas (V). Ejemplo (II)

INDEXADO DE INFORMACIÓN



3. Shards y réplicas (V). Ejemplo (III)

INDEXADO DE INFORMACIÓN



4. Hardware recomendado (I)

Tipos en entorno donde montar nuestro *cluster*.

Testing (Pre-producción)

- Pruebas de componentes, aprendizaje, pruebas, testeo de funcionalidades, instalación, configuraciones...

Producción

- Grandes tasas de IOPS
- Velocidad de procesamiento requerida
- Concurrencia de usuarios

4. Hardware recomendado (II). RAM y CPU

RAM

- Ordenado y operaciones = gran consumo de RAM
- No recomendado el uso de SWAP
- No menos de 8 GiB, no más de 64 GiB (Limitación JVM)

CPU

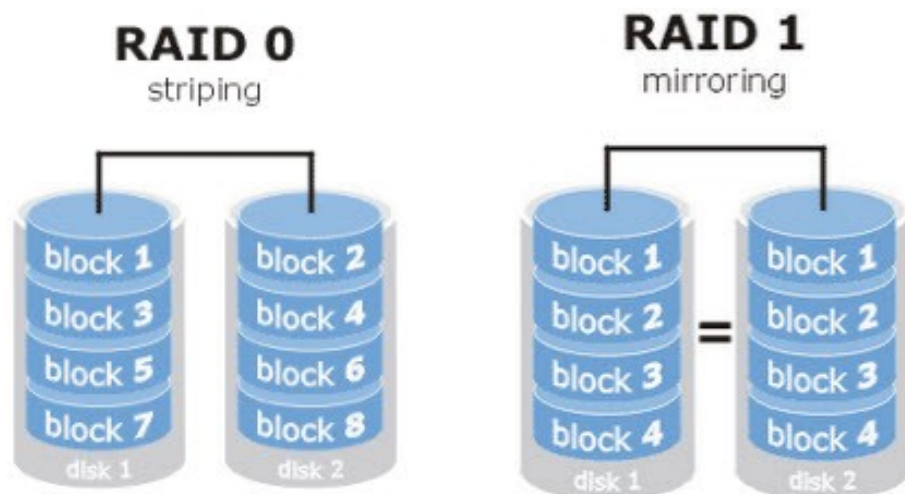
- *Elasticsearch* no requiere excesiva CPU
- Mejor más nodos, núcleos o CPUS que CPUs más rápidas o potentes.

4. Hardware recomendado (III). RAM (JVM)

- **Lucene**: librería *open source* desarrollada en Java para indexado y búsqueda de información.
- Funcionamiento mediante **Java Virtual Machine**.
- *Elasticsearch* se auto configura y se recomienda no tocar la configuración automática (GC, Threads...)
- RAM: 50% *Elasticsearch Heap memory* || 50% libre.
- Máximo 32GiB destinados a *Elasticsearch*.

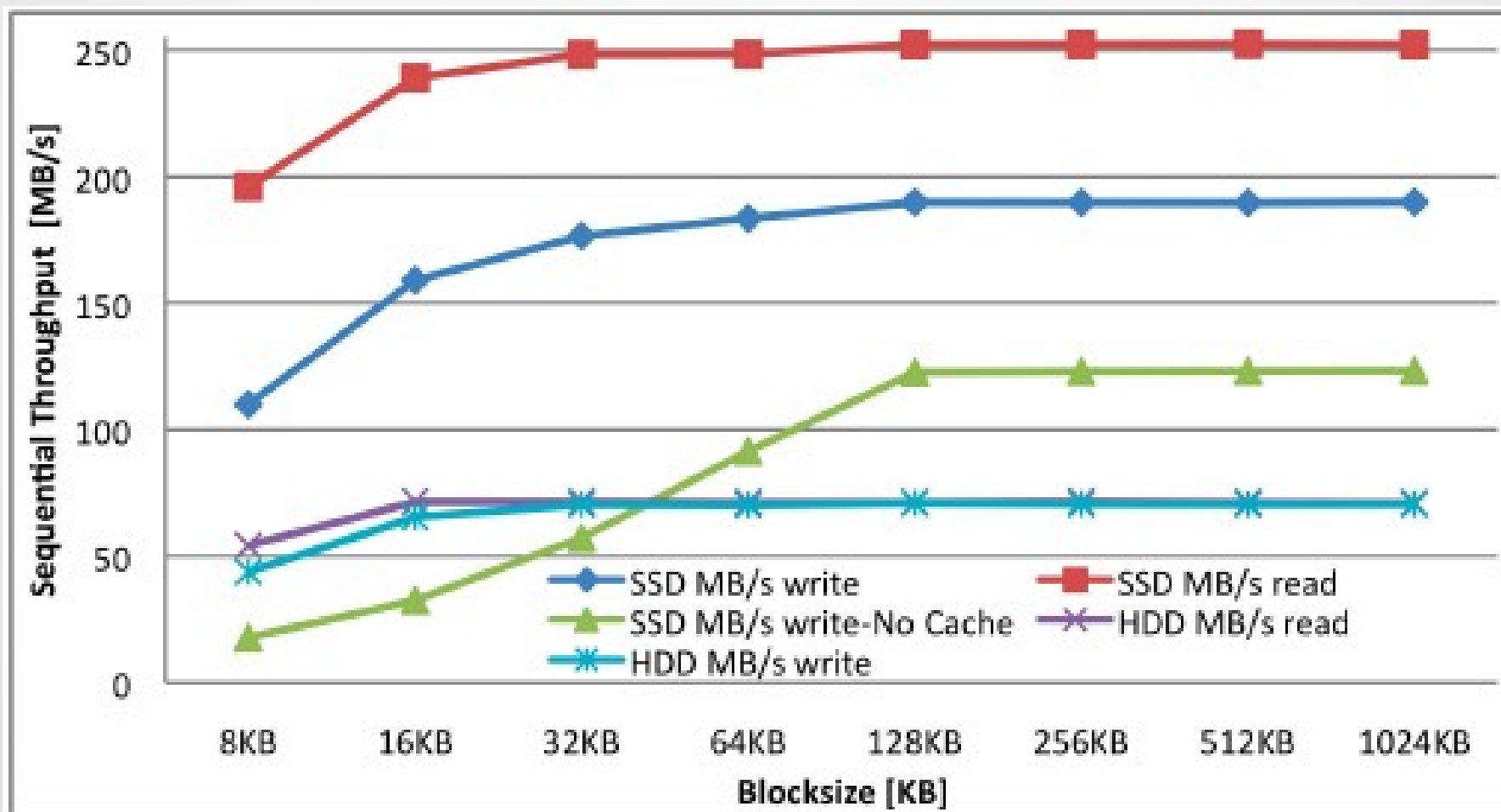
4. Hardware recomendado (IV). IOPS

- Es el **cuello de botella** actual en la mayor parte de sistemas distribuidos.
- Una gran tasa de escritura puede **saturar** los discos.
- Fundamental conseguir las mayores tasas de **IOPS**.
- Recomendado uso de **RAID 0**.



4. Hardware recomendado (V). IOPS

- Recomendado el uso de unidades SSD.



4. Hardware recomendado (VI). Recomendaciones

- No usar almacenamiento en red (**NAS**)
 - Si lo hiciéramos tendríamos altas latencias.
- Tecnología de **red**: 1Gbps o 10Gbps
- ***File descriptors***: *Lucene* usa una gran cantidad de archivos y *sockets*.
 - Necesario incrementar el número de descriptores de archivo a +64.000
 - `cat /proc/sys/fs/file-max`

Webgraphy

- www.elastic.co/
- www.udemy.com/
- duckduckgo.com/ images
- www.researchgate.net
-