Conversación con Gemini

Exported: 15/6/2026 10:08:59

Jordi H.

Has dicho estoy usando logstash. he puesto 3 ficheros conf con "encargos" para guardar 3 ficheros csv con estructuras distintas en elastic en 3 indices distintos. mi sorpresa ha sido que observando el contenido de los nuevos indices, los tres son iguales y tienen mezclados los campos y los contenidos. qué he hecho ma

Gemini

Entiendo perfectamente tu frustración. Es una de las "sorpresas" más comunes cuando uno empieza con Logstash. Lo que estás experimentando no es un error de código per se, sino un malentendido de cómo Logstash gestiona los archivos de configuració

El problema: El "Pipeline" Único

Por defecto, cuando pones varios archivos .conf en la carpeta de configuración, Logstash no los trata como procesos independientes. Lo que hace es **concatenarlos todos en un único pipeline gigante**.Esto significa que

1. Todos los inputs reciben todos los datos.
2. Todos los filtros se aplican a cada evento que entra, sin importar de qué archivo vengan.
3. Todos los outputs envían cada evento a los tres índices a la vez.

Si el Archivo A dice "guarda en índice A" y el Archivo B dice "guarda en índice B", Logstash interpreta: "Para cada evento que pase por aquí, envíalo a A y también a B". Por eso ves los campos mezclados y los datos duplicados en todos los índice

La Solución 1: Pipelines Múltiples (Recomendado)

La forma más limpia y profesional de arreglar esto es usar el archivo pipelines.yml (ubicado generalmente en /etc/logstash/ o en la carpeta config de tu instalación).Debes definir cada "encargo" de forma independiente:YA

```
CODE
# config/pipelines.yml
- pipeline.id: csv_ventas
path.config: "/ruta/a/tus/configs/ventas.conf"

- pipeline.id: csv_usuarios
path.config: "/ruta/a/tus/configs/usuarios.conf"

- pipeline.id: csv_logs
path.config: "/ruta/a/tus/configs/logs.conf"
```

Al hacer esto, Logstash crea "muros" lógicos entre cada configuración. Los filtros de uno no afectarán al otro.

La Solución 2: Etiquetas (Tags) o Condicionales

Si prefieres mantener un solo pipeline, tienes que decirle a Logstash qué eventos pertenecen a cada sitio usando etiquetas en el input y condicionales en el filter y output.**Ejemplo de cómo debería verse u**

CODE

```
input {
file {
path => "/ruta/datos_a.csv"
type => "tipo_a" # Usamos una marca
}
}

filter {
if [type] == "tipo_a" {
# Aquí tus filtros específicos para el archivo A
csv { ... }
}
}

output {
if [type] == "tipo_a" {
elasticsearch {
index => "indice_a"
hosts => ["localhost:9200"]
}
}
}
```

Resumen de acción

- Si los procesos no tienen nada que ver entre sí: Usa la Solución 1 (pipelines.yml). Es más eficiente y fácil de mantener
- Si quieres que todo corra bajo el mismo proceso: Usa la Solución 2 (Condicionales), pero asegúrate de envolver tanto tus filtros como tus outputs en bloques if

¿Te gustaría que te ayude a redactar el archivo pipelines.yml basándome en las rutas reales de tus ficheros?