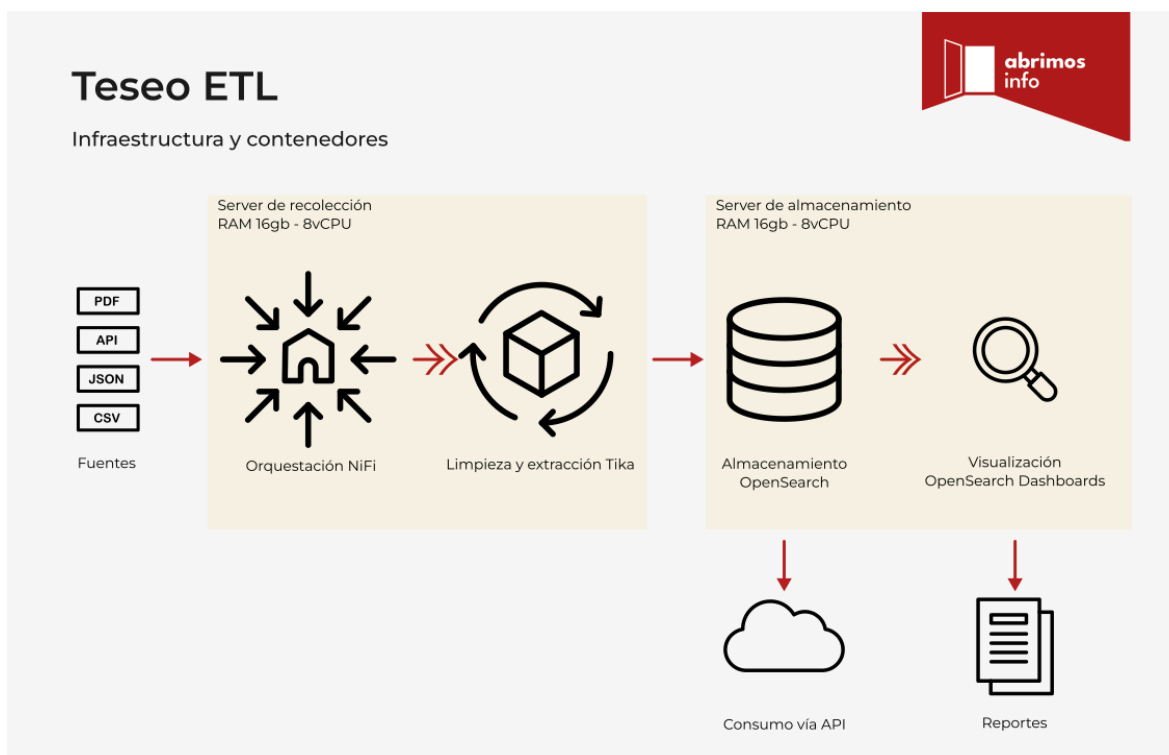


Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

Descripción de la infraestructura



Seguridad

La continuidad de operación y la integridad de los datos y de los servidores es un objetivo principal de TeseoETL.

Todos los componentes se comunican con conexiones cifradas, tanto interna como externamente y todas las interfaces públicas sólo se acceden con usuario y contraseña. Los servicios que no son públicos no están expuestos a internet, reduciendo la superficie de ataque.

Para evitar que se pueda identificar el origen de las solicitudes, se recomienda realizarlas en un servidor independiente con su propia IP. Ya que los procesos de consumo de datos intensivos del servidor de recolección pueden ser detectados y limitados por algunas fuentes de origen. Los usuarios que consuman la API y la visualización desde el servidor de almacenamiento deben estar aislados del servidor de recolección de datos.

Almacenamiento

El data warehouse almacena los datos transformados en un volumen persistente y de configuración flexible. Recomendamos alojar TeseoETL en servidores con almacenamiento seguro y múltiples respaldos.

Los registros primarios (fuentes), intermedios, y logs se almacenan en su propio volumen accesible para el orquestador.

Los archivos para extracción de texto se almacenan en un volumen compartido entre el orquestador y el procesador de extracción.

Los scripts de datos viven en su propio volumen accesible desde el orquestador.

Performance

TeseoETL puede operar múltiples procesos en simultáneo y está programado para requerir cantidades moderadas de memoria, aún cuando se opere con grandes volúmenes de datos. Las consultas son muy rápidas (en general menos de 1 segundo sobre múltiples gigabytes).

El orquestador NiFi permite la configuración de cantidades de procesos simultáneos, y su uso de memoria máximo se define desde la creación del contenedor como parámetro de Java. Aunque 2GB son suficientes para la operación básica de NiFi, recomendamos 8GB ya que los scripts de limpieza y transformación de datos también se ejecutan en este contenedor.

Los scripts de limpieza y transformación, por lo general, tienen un uso constante de memoria por su naturaleza de procesamiento secuencial. Sin embargo, en casos excepcionales es necesario procesar un dataset completo en memoria, lo cual debe ser considerado como adicional al mínimo recomendado del orquestador y especificado mediante el parámetro `-max-old-space-size` de cada script.

El procesador de documentos tiene capacidad de hasta 4 procesos simultáneos de extracción con un límite de memoria recomendado de 4GB.

Requerimientos

Se requiere un sistema capaz de ejecutar contenedores de Docker (docker compose, kubernetes, portainer, etc) y operar almacenamientos de red (NFS).

Se requiere tener dos servidores VPS, cada uno con su propia IP, uno para recolección y otro para almacenamiento.

Cada componente es un contenedor de Docker. Cada servidor tiene algunos componentes.

- Server 1: Recolección.
 - Tamaño recomendado: 8vCPUs, 16gb ram, 100gb disco.
 - Orquestador:: 8GB memoria + 20GB disco
 - Procesador de documentos: 4GB memoria + 1GB disco
- Server 2: Almacenamiento
 - Tamaño recomendado: 8vCPUs, 16gb ram, 100gb disco.
 - Data Warehouse: Recomendado: 4GB memoria + 50GB disco.
 - Nota: El espacio en disco asignado al datawarehouse será el doble de la capacidad total de almacenamiento del sistema, ya que estos datos se duplican para garantizar alta disponibilidad.
 - Business Intelligence: Recomendado: 2GB memoria + 1GB disco

Dominios

Es necesario contar con un dominio principal y subdominios para cada servicio debido a la arquitectura distribuida de TeseoETL. Cada componente (NiFi, OpenSearch y Dashboards) cumple una función específica y requiere configuraciones de red independientes para asegurar su correcto funcionamiento y seguridad.

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

- **Aislamiento de servicios:** Al utilizar subdominios separados, cada servicio opera de manera independiente, lo que facilita la administración, el monitoreo y la resolución de problemas.
- **Seguridad y certificados SSL:** Los subdominios permiten implementar certificados SSL individuales mediante Certbot, asegurando conexiones seguras y cifradas para cada componente.
- **Escalabilidad y mantenimiento:** Esta configuración facilita la escalabilidad y el mantenimiento de los servicios, permitiendo actualizaciones o modificaciones sin afectar el funcionamiento de otros componentes.
- **Acceso especializado:** Cada subdominio permite un acceso directo y especializado a las interfaces de los servicios, mejorando la experiencia del usuario y la integración con otras herramientas.

Volúmenes

El almacenamiento en TeseoETL se maneja a través de volúmenes montados en los contenedores de Docker. Se dividen en dos categorías:

- **Volúmenes mapeados desde el repositorio:**
 - Contienen configuraciones clave, scripts y definiciones de pipelines de NiFi.
 - Se almacenan en Git para mantener un historial de cambios y facilitar la replicación en otros entornos.
 - No se deben modificar directamente en producción sin hacer un commit previo en el repositorio.
- **Volúmenes de almacenamiento:**

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

- Almacenan datos transformados, logs de ejecución y archivos intermedios utilizados por NiFi y OpenSearch.
- Pueden configurarse como volúmenes **NFS (Network File System)** para permitir la escalabilidad y la redundancia en múltiples servidores.
- Si se opta por volúmenes locales en los servidores, se debe considerar que esto limitará la expansión del clúster de OpenSearch, ya que cada nodo solo podrá acceder a su propio almacenamiento.

Los volúmenes que ocupan mucho espacio son el `provenance_repository` y el `flowfile_repository` de NiFi y los volúmenes de datos de OpenSearch. Estos son los que únicos que varían significativamente de acuerdo a la cantidad de datos que se procesen y almacenen.

Instalación Paso a paso

Requisitos

- **Git** (con llave SSH) para acceder al repositorio o una **copia del código** en zip.
- O **unzip** si se usan archivos zip para distribuir el código.

TeseoETL ha sido probado en Ubuntu 24.04, pero debería funcionar en otras distribuciones también.

Instalación mediante Docker Compose con Nginx

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

1. Clonar el repositorio o copiar el zip del código en ambos servidores:

Usando git:

```
apt install git
```

```
git clone git@github.com:Abrimos-info/teseoetl.git
```

[cambiar por la dirección del repositorio correspondiente a cada cliente]

```
cd teseoetl
```

Usando zip:

```
scp teseoel-main.zip root@[IP]:/root
```

```
ssh root@[ip]
```

```
apt update && apt install unzip
```

```
unzip teseoetl-main.zip
```

```
cd teseoetl-main
```

2. Instalar requerimientos en ambos servidores:

```
./install-teseoetl.sh
```

3. Configurar acceso web:

Debes configurar tres subdominios en tu servidor DNS. Crear entradas de tipo A con la dirección IP de cada servidor y los siguientes hostnames:

- **nifi.[TUDOMINIO]** apuntando al **Servidor 1**.
- **opensearch.[TUDOMINIO]** apuntando al **Servidor 2**.
- **dashboards.[TUDOMINIO]** apuntando al **Servidor 2**.

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

4. Configurar Nginx en ambos servidores:

Copia y edita los templates de configuración de nginx de la carpeta nginx del repositorio a `/etc/nginx/sites-enabled/`

Servidor 1:

```
cp ./nginx/sites-enabled/nifi.example  
/etc/nginx/sites-enabled/nifi.[TUDOMINIO].conf
```

Servidor 2:

```
cp ./nginx/sites-enabled/dashboards.example  
/etc/nginx/sites-enabled/dashboards.[TUDOMINIO].conf
```

```
cp ./nginx/sites-enabled/opensearch.example  
/etc/nginx/sites-enabled/opensearch.[TUDOMINIO].conf
```

Ejemplo de configuración:

```
server {  
    server_name [SUBDOMINIO].[TUDOMINIO];  
    root /var/www/html;  
  
    location / {  
        proxy_pass https://localhost:[PUERTO]/;  
        proxy_cookie_domain localhost [SUBDOMINIO].[TUDOMINIO];  
        proxy_set_header X-ProxyHost [SUBDOMINIO].[TUDOMINIO];  
        proxy_set_header X-ProxyPort 443;  
    }  
  
    client_max_body_size 100M;
```

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

}

- **[PUERTO]** debe ser:
 - 5601 o el valor de .env para \$OPENSEARCH_DASHBOARDS_PORT para **dashboards**
 - 9200 para **opensearch** o el valor de OPENSEARCH_NODE1_PORT en .env
 - 8443 para **nifi** o el valor de NIFI_PORT en .env
- **[SUBDOMINIO]** puede ser **nifi**, **opensearch** o **dashboards** (o cualquier otro que prefieras, sólo recuerda actualizarlo en .env).
- **[TUDOMINIO]** es el dominio o subdominio que uses, por ejemplo:
teseoetl.tuorganizacion.com.

5. Ejecutar Certbot:

Después de configurar los dominios y Nginx, ejecuta Certbot para crear los certificados SSL en cada servidor:

certbot

Poner una dirección de correo electrónico para recibir notificaciones, aceptar el eula, negar compartir y seleccionar los dominios, si son dos, se pone "1 2".

Sigue las instrucciones para confirmar los dominios y asegúrate de activar la redirección a HTTPS.

6. Configurar Docker Compose en cada servidor:

- En el **Servidor 1**:

Editar el archivo .env y cambiar el nombre de dominio en la variable
NIFI_HOSTNAME

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

Si se ha cambiado las configuraciones de hostname y password de opensearch, también actualizarlas aquí.

Adicionalmente se recomienda cambiar la contraseña NIFI_SINGLE_USER_PASSWORD y opcionalmente también NIFI_TRUSTSTORE_PASSWORD.

Otros parámetros a configurar son:

- Los límites de CPU y memoria de acuerdo a la capacidad del servidor.
- La afinidad del deploy (en qué servidores puede ubicarse cada contenedor).
- Puertos. El puerto externo puede modificarse si hubiera conflictos en el servidor o si se quieren correr dos instancias de teseoetl.

Para esta configuración no se usarán las networks, entonces es necesario editar el archivo docker-compose-nifinode.yml para desactivar las redes, esto es poner un # al inicio de la línea en la clave networks y en las claves en services.*.networks. Estas claves sólo sirven si se usará docker swarm.

Ejecutar:

```
docker compose -f docker-compose-nifi-tika.yml up -d
```

Nota: Si aparece un error en docker sobre que no se tiene el comando compose, es necesario instalar docker-compose-plugin. No usar el comando usar docker-compose (con guión).

- En el **Servidor 2**:

Editar el archivo .env

- OPENSEARCH_HOSTNAME con el dominio utilizado para opensearch.

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

- OPENSEARCH_PASSWORD cambiando la contraseña por defecto

Otros parámetros a configurar son:

- Los límites de CPU y memoria de acuerdo a la capacidad del servidor.
- La afinidad del deploy (en qué servidores puede ubicarse cada contenedor).
- Puertos. El puerto externo puede modificarse si hubiera conflictos en el servidor o si se quieren correr dos instancias de teseoetl.

Nota: No modificar username y password para OpenSearch en esta etapa, podemos hacerlo más adelante cuando ya tengamos acceso a la herramienta.

Recordar que estos valores también deben ser modificados en el .env del servidor 1 para las conexiones desde NiFi.

Antes de dar de alta el docker, nos aseguramos que las carpetas de datos tienen los permisos correctos:

```
sudo chown -R 1000:1000 ./opensearch/data
```

Para esta configuración no se usarán las networks, entonces es necesario editar el archivo docker-compose-nifinode.yml para desactivar las redes, esto es poner un # al inicio de la línea en la clave networks y en las claves en services.*.networks. Estas claves sólo sirven si se usará docker swarm.

Ejecutar:

```
docker compose -f docker-compose-opensearch.yml up -d
```

Verificar el estado de los servicios:

Los servidores pueden tardar varios minutos en quedar completamente funcionales, sobre todo en la primera corrida, que se requiere descargar los contenedores.

Usa los siguientes comandos para asegurarte de que todo está funcionando correctamente:

```
docker ps
```

```
docker logs
```

Con esto, TeseoETL debería estar funcionando y listo para su configuración y uso. Si al final de los logs aparecen mensajes de error y el servidor no funciona, revisar en detalle en la sección "Problemas comunes" más abajo.

Acceder a nifi, opensearch y dashboards con los dominios y password configurados más arriba.

Configuraciones iniciales de Opensearch Dashboards

Para garantizar la experiencia adecuada y uniforme de TeseoETL, luego de la instalación y configuración se necesitan hacer algunas modificaciones manualmente en Opensearch Dashboards.

Diríjase a la configuración avanzada de OpenSearch, ej: <https://172.20.0.1:5601/app/settings>

Busque la opción Theme Version y modifique el selector a la version 9 (preview).

Gestión de usuarios

El control de acceso a TeseoETL varía según el componente del sistema:

- **NiFi:** Actualmente, NiFi utiliza un único usuario que se configura mediante variables de entorno en el archivo `docker-container-nifi-tika.yml` (`services.nifi.environment`), a su vez estas se toman del archivo `.env`

En están:

```
- NIFI_SINGLE_USER_PASSWORD  
  
- NIFI_SINGLE_USER_USERNAME
```

Esto implica que todas las operaciones dentro de NiFi se realizan bajo el mismo usuario, sin una gestión granular de permisos. Para mejorar la seguridad:

- Configurar las credenciales de acceso en el archivo de variables de entorno antes del despliegue.
- Restringir el acceso a NiFi mediante reglas de firewall o redes privadas virtuales (VPN).
- Implementar autenticación basada en certificados si se requiere mayor seguridad.

Para más información sobre configuración de usuarios en NiFi por favor revisar el manual oficial:

https://nifi.apache.org/nifi-docs/administration-guide.html#single_user_identity_provider

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

- **OpenSearch y OpenSearch Dashboards:**
 - La gestión de usuarios en OpenSearch se realiza mediante el módulo de seguridad integrado. Los administradores pueden crear múltiples usuarios con diferentes privilegios.
 - **Roles:** Se deben definir roles con permisos específicos para restringir el acceso a ciertas áreas del sistema, como la visualización, indexación o eliminación de datos.
 - Cómo crear usuarios y roles
<https://opensearch.org/docs/2.19/security/access-control/users-roles/#opensearch-dashboards>
 - Para acceder a los index_patterns los usuarios deben tener el cluster role llamado **cluster_composite_ops_ro**
 - Para poder ver las configuraciones aplicadas deben poder tener acceso read a **.kibana**
 - Asegúrate de mapear los usuarios al role después de haberlo creado.
 - **Workspaces:** Se pueden configurar espacios de trabajo personalizados dentro de OpenSearch Dashboards para que cada equipo de usuarios tenga acceso solo a los índices de datos relevantes para sus necesidades.
 - Cómo crear un espacio de trabajo:
<https://opensearch.org/docs/2.19/dashboards/workspace/create-workspace/>

Gestión de acceso por índices y columnas

Para cada rol de usuario se puede definir si tienen acceso a cada índice y también se puede excluir o anonimizar los valores por columna.

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

Al crear el rol se le agrega un permiso de índice para cada índice al que puede acceder el usuario con este rol. En este se le otorgan los permisos de get y de search. Además se puede configurar columnas excluidas y anonimizadas.

Ejemplo de un rol, vista de edición, que tiene los permisos de cluster correctos y los permisos de índice incluyendo exclusión y anonimización de valores.

Cluster Permissions
Specify permissions using either action groups or single permissions. An action group is a list of single permissions. You can also create your own reusable permission groups.

cluster_composite_ops_ro ×

Create new permission group

Index permissions
Index permissions allow you to specify how users in this role can access the specific indices. By default, no index permissions are granted. [Learn more](#)
> guatecompras_ocds, .kibana

test_index

Index
test_index ×
Specify index pattern using *

Index permissions
You can specify permissions using both action groups or single permissions. A permission group is a list of single permissions. You can also create your own reusable permission groups.

get × search ×

Create new permission group

Document level security - optional
You can restrict a role to a subset of documents in an index. [Learn more](#)

```
{
  "bool": {
    "must": {
      "term": {
        "type": "text"
      }
    }
  }
}
```

Field level security - optional
You can restrict what document fields a user can see. If you use field-level security in conjunction with document-level security, you must specify the field that document-level security uses.

Exclude ×

columna_b ×

Anonymization - optional
Masks any sensitive fields with a random value to protect your data security.

columna_a ×

Dataviews y dashboards iniciales

Si hay ejemplos que queremos incorporar, se pueden importar los saved objects. Sino hay que crear a mano las dataviews y dashboards que queremos que los usuarios puedan acceder en su primera visita.

Respaldos y mudanzas

Para garantizar la integridad de los datos y la configuración del sistema, se deben implementar estrategias de respaldo periódicas:

1. Configuraciones y scripts en Git:

- Mantener versionados todos los scripts, definiciones de pipelines y configuraciones dentro de un repositorio Git.
- Realizar commits regulares después de cada cambio importante.
- En caso de migración, clonar el repositorio en el nuevo entorno y desplegar las configuraciones.

2. Volúmenes de datos:

- Realizar copias de seguridad de los volúmenes de almacenamiento mediante herramientas como **rsync** o **tar**.
- Programar respaldos automáticos en almacenamiento externo para evitar pérdida de datos.

3. Instantáneas de OpenSearch:

- Configurar la funcionalidad de snapshots en OpenSearch para respaldar índices completos.
- Guardar estos snapshots en un sistema de almacenamiento externo.
- Utilizar snapshots para restaurar índices en caso de corrupción o eliminación accidental.

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

4. Respaldos de configuración de NiFi:

- Los flujos de NiFi están en el volumen `./nifi/conf`, en particular el archivo `flow.json.gz`
- Se pueden comitear los cambios a este archivo para tener un respaldo del mismo.

Procedimiento recomendado para migraciones

Si fuera necesario mover los archivos a otro servidor, ten en cuenta las siguientes consideraciones:

- **Antes de la migración:**

1. Realizar un respaldo completo de las configuraciones en Git.
2. Exportar instantáneas de los datos almacenados en OpenSearch.
3. Guardar copias de los volúmenes de almacenamiento en un servidor externo.

- **Durante la migración:**

1. Desplegar la infraestructura en el nuevo entorno, siguiendo las instrucciones de este documento.
2. Restaurar las configuraciones desde Git.
3. Importar las instantáneas de OpenSearch.
4. Montar los volúmenes de almacenamiento restaurados.

- **Después de la migración:**

1. Verificar la funcionalidad de todos los servicios.
2. Ajustar configuraciones y permisos según sea necesario.
3. Implementar una política de respaldo continua en el nuevo entorno.

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

Con esta estrategia, se garantiza la continuidad operativa de TeseoETL, reduciendo el riesgo de pérdida de datos y facilitando la expansión del sistema en el futuro.

Variables de entorno y el archivo .env

En TeseoETL hay tres tipos de variables de entorno:

- Las que están en el archivo .env, que están sólo disponibles para los docker-compose (salvo que este las exponga al servicio también).
- Las que están definidas en los docker-compose, que pueden venir también del .env y están disponibles para cada servicio.
- Las que son definidas por cada servicio internamente en su contenedor y pueden depender de su configuración o comportamientos por defecto.

El archivo .env define algunas configuraciones secretas que no deben registrarse en un repositorio de git. Estas configuraciones también podrían guardarse en el mismo portainer.

Variables de entorno de OpenSearch

Nombre	Valor inicial	Descripción
OPENSEARCH__INITIAL__ADMIN__PASSWORD	adminpass	Default admin password for OpenSearch cluster on first start
OPENSEARCH__USERNAME	admin	Username for OpenSearch authentication
OPENSEARCH__PASSWORD	\${OPENSEARCH__INITIAL__ADMIN__PASSWORD}	Password for OpenSearch authentication (references the admin password defined above, but

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

	(toma el valor de la variable de arriba, pero se puede cambiar)	can be changed once there is a user for nifi)
OPENSEARCH_JAVA_OPTS	-Xms4g -Xmx4g	Java options for OpenSearch, setting minimum and maximum heap size to 4GB
OPENSEARCH_CLUSTER_NAME	opensearch-cluster	Name of the OpenSearch cluster
OPENSEARCH_NODE1_NAME	opensearch-node1	Name for the first OpenSearch node. Esto también se ve en el nombre del contenedor de docker, y en el hostname.
OPENSEARCH_NODE2_NAME	opensearch-node2	Name for the second OpenSearch node. Esto también se ve en el nombre del contenedor de docker, y en el hostname.
OPENSEARCH_NODE1_PORT	9200	HTTP port for the first OpenSearch node
OPENSEARCH_NODE2_PORT	9201	HTTP port for the second OpenSearch node
OPENSEARCH_DASHBOARDS_PORT	5601	Port for OpenSearch Dashboards web interface. Esto se usa en la configuración de nginx.
OPENSEARCH_MEMORY_LIMIT	6G	Memory limit for OpenSearch nodes. Esto lo limita a nivel docker.
OPENSEARCH_DASHBOARDS_MEMORY_LIMIT	4G	Memory limit for OpenSearch Dashboards

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

OPENSEARCH_NODE1_PLACEMENT_ROLE	worker	Placement role for node1 in the cluster (for container orchestration)
OPENSEARCH_NODE1_PLACEMENT_HOSTNAME	teseodev2	Hostname for node1 placement
OPENSEARCH_NODE2_PLACEMENT_ROLE	worker	Placement role for node2 in the cluster (for container orchestration)
OPENSEARCH_NODE2_PLACEMENT_HOSTNAME	teseodev2	Hostname for node2 placement
OPENSEARCH_DATA1_DIR	./opensearch/data/1	Data directory for OpenSearch node1
OPENSEARCH_DATA2_DIR	./opensearch/data/2	Data directory for OpenSearch node2

Variables de entorno de NiFi

Nombre	Valor inicial	Descripción
NIFI_PORT	8443	HTTPS port for Apache NiFi web interface. Esto se usa en la configuración de nginx.
NIFI_HOSTNAME	nifi.teseodev1.abrimos.info	Hostname for NiFi service
NIFI_SINGLE_USER_USERNAME	admin	Username for NiFi single user authentication
NIFI_SINGLE_USER_PASSWORD	nifipass	Password for NiFi single user authentication
NIFI_TRUSTSTORE_PASSWORD	changeme-teseoetl-default-truststore-password!	Password for NiFi's truststore (for SSL/TLS configuration)

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

NIFI_SENSITIVE_PROPS_KEY	teseoetl-sensitive-prop s-key	Key used to encrypt sensitive properties in NiFi
NIFI_MEMORY_LIMIT	8G	Memory limit for NiFi service. Se aplica desde docker.
NIFI_FLOWFILE_REPOSITORY _DIR	./nifi/flowfile_repositor y	Directory for NiFi flowfile repository
NIFI_PROVENANCE_REPOSIT ORY_DIR	./nifi/provenance_repo sitory	Directory for NiFi provenance repository
OPENSEARCH_PUBLIC_PORT	443	Public HTTP port for the main OpenSearch node (via nginx)
OPENSEARCH_HOSTNAME	opensearch.teseodev2. abrimos.info	Hostname for the main opensearch node (via nginx)
OPENSEARCH_URL	OPENSEARCH_URL=http s://\${OPENSEARCH_U SERNAME}:\${OPENSEAR CH_PASSWORD}@\${OP ENSEARCH_HOSTNAME }:\${OPENSEARCH_PUBL IC_PORT}	Composed OpenSearch URL for NiFi. Complete URL for NiFi to connect to OpenSearch, including authentication.

Variables de entorno de Tika

TIKA_PORT	9998	Port for Apache Tika server (used for document parsing)
TIKA_MEMORY_LIMIT	8G	Memory limit for Tika service

Problemas comunes

Se acabó el espacio en disco

Síntoma: Los servicios dejan de funcionar o presentan errores de escritura.

Solución:

- Verificar el uso de disco con `df -h`.
- Liberar espacio eliminando archivos temporales o registros antiguos.
- Considerar ampliar la capacidad de almacenamiento si es un problema recurrente.

Se reinició OpenSearch

Síntoma: El servicio de OpenSearch no está disponible o se reinicia constantemente.

Solución:

- Verificar los registros con `docker logs [nombre_del_contenedor]`.
- Asegurarse de que hay suficiente memoria disponible.
- Reiniciar manualmente el servicio si es necesario:

```
docker restart [nombre_del_contenedor]
```

En ocasiones puede ser necesario hacer u ciclo de down/up, así:

```
docker compose -f docker-compose-opensearch.yml down
```

```
docker compose -f docker-compose-opensearch.yml up -d
```

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

Si crees que el problema está en los volúmenes agrega un -v a el down.

Revisar logs de los servicios

Síntoma: Problemas generales en la ejecución de los servicios.

Solución:

- Usar `docker logs [nombre_del_contenedor]` para obtener detalles de los errores.
- Revisar logs específicos en las rutas configuradas para cada servicio.

Los scripts `teseo_scripts` ejecutados por NiFi saturan la memoria

Síntoma: El sistema se vuelve lento o deja de responder durante la ejecución de scripts.

Solución:

- Optimizar los scripts para reducir el consumo de memoria.
- Incrementar la memoria disponible si es posible.
- Configurar límites de uso de memoria en NiFi para evitar saturaciones.

Los scripts de NiFi dan error por dependencias desactualizadas

Síntoma: Errores de ejecución relacionados con versiones de librerías o paquetes.

Solución:

Documentación interna

Manual de uso TeseoETL - v2.02 - Actualizado 26/03/2025

- Actualizar las dependencias utilizando el gestor de paquetes correspondiente (por ejemplo, [pip](#), [npm](#)).
- Verificar la compatibilidad de las versiones antes de actualizar.
- Documentar las versiones compatibles para futuras referencias.