**Gluster FS**

GlusterFS es un sistema de archivos basado en software libre (GPLv3), escalable y distribuido en red para montar un NAS (Network Attached Storage). Resulta muy útil y flexible para almacenamiento en computación de nube. Fue desarrollado inicialmente por Gluster Inc., compañía adquirida por RedHat en el año 2011.    
   
**1. Rol de GlusterFS en el proyecto**

GlusterFS será el sistema de almacenamiento compartido y replicado entre todos los nodos web del Docker Swarm.

Su función principal aquí será:

* Garantizar que todas las instancias Web vean y utilicen exactamente los mismos archivos.
* Mantener los datos sincronizados entre los nodos para que, aunque un contenedor se mueva o se reinicie en otro servidor, siga teniendo acceso a los mismos datos.
* Soportar tolerancia a fallos: si un nodo con datos cae, el resto mantiene el servicio operativo

**2. Cómo funciona GlusterFS**

Conceptos clave

* Nodo: Cada servidor físico o VM que ejecuta GlusterFS.
* Brick: Carpeta o disco que GlusterFS usa como unidad de almacenamiento dentro de un nodo.
* Volumen: Agrupación de bricks que se comporta como un solo sistema de archivos.
* Cliente: Aquí serían los mismos nodos del Swarm montando el volumen GlusterFS.

Tipo de volumen para el proyecto

Para alta disponibilidad en sitios web escalables, normalmente se usa:

* Replica 3 → Cada archivo se guarda en 3 nodos distintos.
* Esto asegura que, si caen 1 o 2 nodos, los datos siguen estando disponibles.

**3. Flujo de trabajo del proyecto en la parte de almacenamiento**

1. Nodo Web crea/modifica archivo.
2. GlusterFS replica ese archivo en los demás nodos del volumen.
3. Otro nodo Web accede al mismo archivo sin necesidad de sincronizaciones manuales.
4. Si un nodo falla, Docker Swarm migra el contenedor a otro servidor, que seguirá teniendo acceso al mismo almacenamiento.

Arquitectura

Multi-Volume Replicated Storage

Es ideal para entornos distribuidos donde se requiere aislamiento lógico, alta disponibilidad y consistencia entre nodos. Dos volúmenes replicados sobre tres nodos, permite separar funciones (frontend/backend) y mantener resiliencia ante fallos.