Prácticas COS – Curso 2022-23 – Sesión 5
Sistema de Ficheros *GlusterFS*

GlusterFS es un sistema de ficheros distribuido de código abierto que fue adquirido recientemente por Red Hat. Permite agregar varios servidores de archivos sobre Ethernet o Infiniband, permitiendo la creación de un gran (varios Petabytes) sistema de archivos en red. Se pueden replicar y/o distribuir ficheros a través de los distintos servidores de almacenamiento. Es una forma económica de crear un sistema de almacenamiento NAS (Network Attached Storage) utilizando PCs convencionales. Tiene una estructura cliente-servidor.

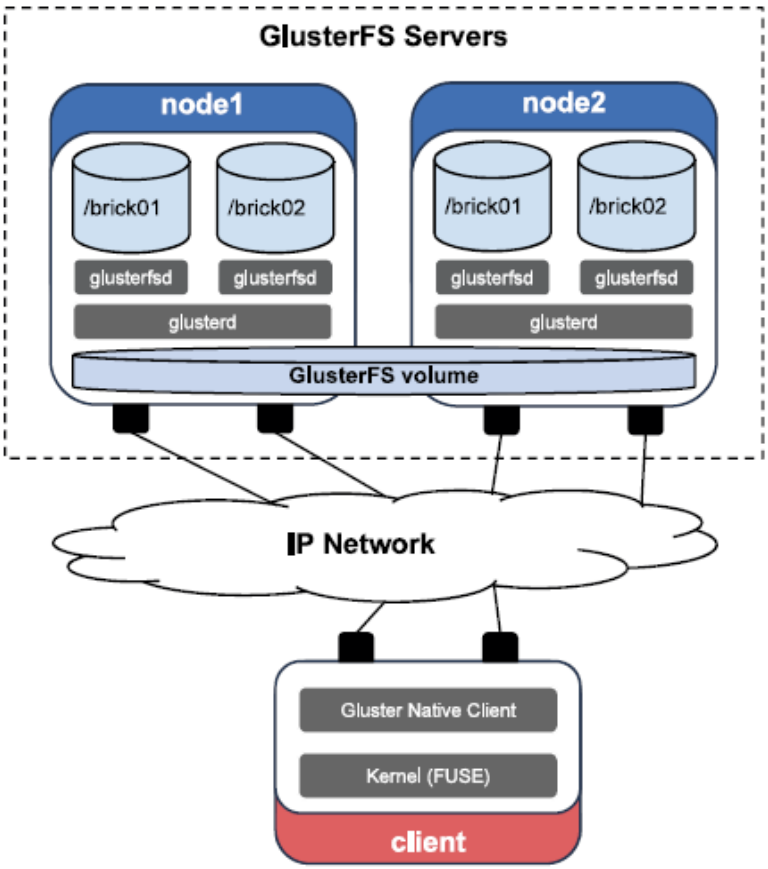


Figura 1. Arquitectura simplificada de un cluster de almacenamiento GlusterFS de dos nodos.

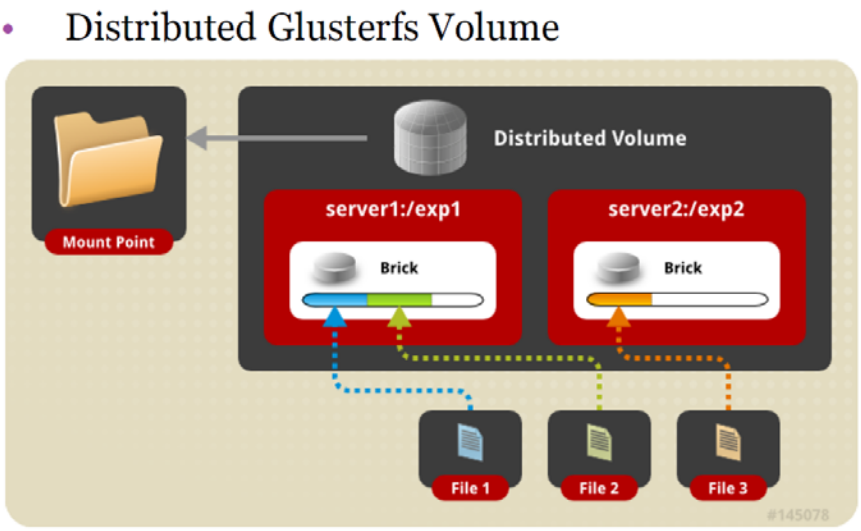
Un **sistema de archivos distribuido** es un sistema de archivos en el que los datos se distribuyen entre varios nodos y los usuarios pueden tener acceso a estos datos sin conocer la ubicación real de los archivos. El usuario no experimenta la sensación de acceso remoto.

En la figura 1 se puede apreciar la estructura simplificada del sistema de ficheros Gluster. Dispone de un conjunto de servidores (servers pool) que exportan uno o varios bricks. Un **brick** es básicamente una partición con un sistema de ficheros local, contiene el nombre de un servidor y un directorio donde los clientes podrán almacenar sus ficheros. Tiene la forma node1:/brick01.

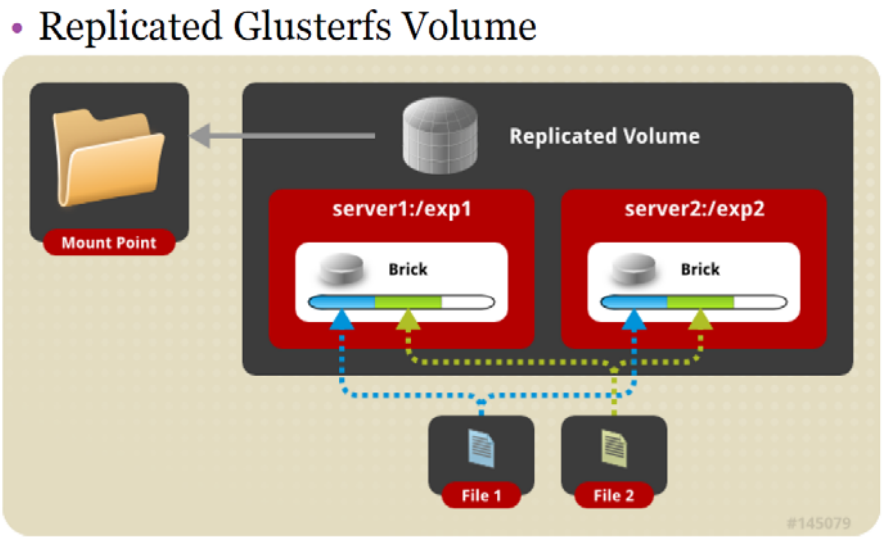
Un **volumen** es una colección de bricks. Los volúmenes se comparten con los clientes a través de CIFS, NFS o Gluster File System. Gluster soporta varios tipos de volúmenes, cada uno con unas características propias en lo referente a la disponibilidad de los datos y escalabilidad.

Los tipos de volumen más utilizados son:

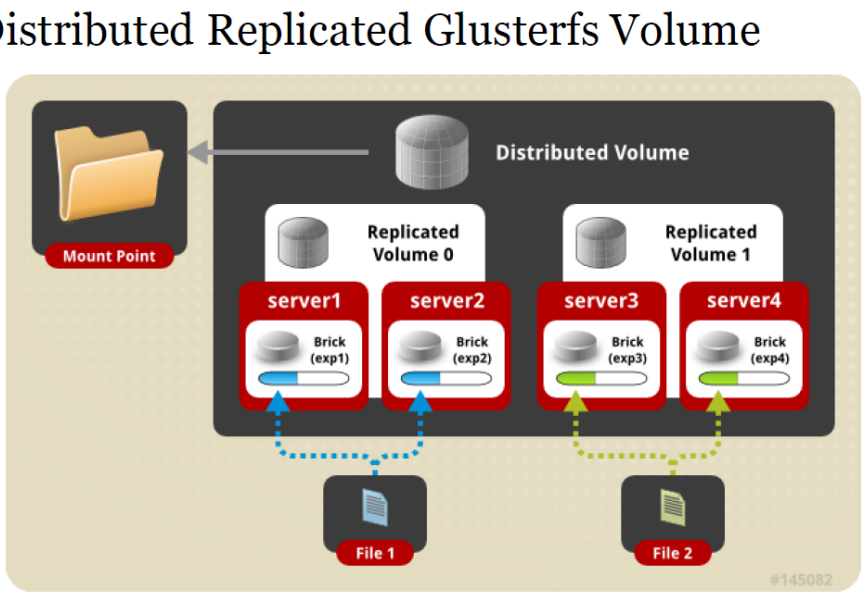
- **Volumen distribuido:** Este es el tipo de volumen por defecto en GlusterFS, si no se especifica un tipo de volumen concreto. Los archivos se distribuyen en varios bricks en el volumen, de forma que el archivo 1 sólo podrá almacenarse en el brick 1 o en el brick 2, pero no en ambos, por lo que no habrá redundancia de datos. Este tipo de volumen distribuido hace que sea más fácil y barato escalar el tamaño del volumen. No obstante, este tipo de volumen, al no proporcionar redundancia, puede sufrir la pérdida de los datos en caso de que uno de los dos bricks falle, por lo que es necesario realizar un backup de los archivos con una aplicación externa a GlusterFS.



- **Volumen replicado:** Con este tipo de volumen eliminamos el problema ante la pérdida de datos que se experimenta con el volumen distribuido. En el volumen replicado se mantiene una copia exacta de los datos en cada uno de los bricks. El número de réplicas se configura por el usuario al crear el volumen, si queremos dos réplicas necesitaremos al menos dos bricks, si queremos tres réplicas necesitaremos tres bricks, y así sucesivamente. Si un brick está dañado, todavía podremos acceder a los datos mediante otro brick. Este tipo de volumen se utiliza para obtener fiabilidad y redundancia de datos.



Volumen distribuido replicado: Combina las características de los dos anteriores. En este tipo de volumen los datos se distribuyen en conjuntos duplicados de bricks. El número de bricks debe ser un múltiplo del número de réplicas. También es importante el orden en que especifiquemos los bricks porque los bricks adyacentes serán réplicas entre ellos. Si tenemos ocho bricks y configuramos una réplica de dos, los dos primeros bricks serán réplicas el uno del otro, y luego los dos siguientes, y así sucesivamente. Esta configuración se denomina 4×2. Si tenemos ocho bricks y configuramos una réplica de cuatro, los cuatro primeros bricks serán réplicas entre ellos y se denominará 2×4.



1. Preparación

Para esta práctica usaremos los 3 nodos servidores que hemos creado en sesiones anteriores, `cluster2`, `cluster3`, y `cluster4`.

Para esta sesión, creamos dos discos nuevos, uno de 250MB (`gfs1.vdi`) y otro de 300 MB (`gfs2.vdi`) y los conectamos al controlador SATA de las máquinas virtuales `cluster3 y cluster4` (esto es, añadimos dos discos a cada máquina). La idea es configurar un sistema de archivos *GlusterFS* en las máquinas `cluster3 y cluster4 (server pool),` y montarlo en el nodo `cluster2` que actuará como cliente.

2. `[cluster3,cluster4]` Preparación de los *bricks*

El primer paso será conocer el nombre que el sistema operativo ha dado a los nuevos dispositivos:

```
# lsblk
```

O también:

```
# cat /proc/partitions
```

Supondremos que los nuevos dispositivos son `/dev/sdb` y `/dev/sdc`, respectivamente. Hay que tener en cuenta que dicho nombre podría cambiar de una máquina a otra, si el número y características de los discos conectados difiere en cada máquina. Una vez conocemos los nombres, procederemos a crear una única partición en cada uno de ellos, con la utilidad *fdisk*.

```
# fdisk /dev/sdc
```

Elegimos *"n"* para crear una nueva partición. No necesitamos especificar nada más, así que para los siguientes datos que pide podemos usar los valores por defecto que ofrece *fdisk*, asi que pulsaremos *"Intro"* hasta finalizar. Para finalizar *"w"* indicando que queremos escribir esta partición en el disco y hacerla efectiva.

Seguidamente, los formatearemos mediante la orden `mkfs`. En este caso, utilizaremos el sistema de archivos xfs. Es posible que tengamos que instalar el paquete correspondiente (xfsprogs).

```
# mkfs.xfs /dev/sdb1
# mkfs.xfs /dev/sdc1
```

Crearemos los puntos de montaje:

```
# mkdir –p /export/brick1
# mkdir –p /export/brick2
```

Y procedemos a montarlos. En este caso, cada máquina ofrecerá dos *bricks*, donde cada uno de ellos estará asociado con uno de los nuevos dispositivos. En particular:

```
# mount /dev/sdb1 /export/brick1
# mount /dev/sdc1 /export/brick2
```

Por descontado, si deseáramos consolidar la configuración, deberíamos crear las entradas correspondientes en el archivo `/etc/fstab`.

Una vez montado, creamos el subdirectorio que contendrá los datos:

```
# mkdir /export/brick1/data
# mkdir /export/brick2/data
```

En este momento, cada máquina ofrece sendos *bricks* en las rutas `/export/brick1/data` y `/export/brick2/data`

Instalaremos los paquetes necesarios en ambos servidores:

```
# yum install –y centos-release-gluster
# yum install –y glusterfs-server
# systemctl start glusterd
# systemctl enable glusterd
# glusterfs --version
```

3. [cluster3] Configuración de *GlusterFS*

Configuraremos el grupo de nodos, lanzando la siguiente orden desde uno de los nodos que componen el grupo (por ejemplo desde `cluster3`):

```
# gluster peer probe cluster4
```

Podemos comprobar el estado del grupo mediante la orden (por ejemplo desde `cluster3`):

```
# gluster peer status
```

4. [cluster3] Preparación de un volumen replicado.

Desde uno de los nodos del grupo (por ejemplo `cluster3`), crearemos un volumen replicado con los discos de 250MB mediante la orden:

```
# gluster volume create gv0 replica 2 cluster3:/export/brick1/data \
```

```
      cluster4:/export/brick1/data
```

Y lo iniciamos con:

```
# gluster volume start gv0
```

Podemos obtener las características del volumen con:

```
# gluster volume info
```

5. [cluster3] Preparación de un volumen distribuido.

Desde uno de los nodos del grupo (por ejemplo `cluster3`), crearemos un volumen distribuido con los discos de 300MB mediante la órden:

```
# gluster volume create gv1 cluster3:/export/brick2/data  \
      cluster4:/export/brick2/data
```

Y lo iniciamos con:

```
# gluster volume start gv1
```

Podemos obtener las características de los volumenes disponibles con:

```
# gluster volume info
```

Por seguridad es posible limitar el acceso los volúmenes a máquinas con determinadas direcciones de red. Por ejemplo, vamos permitir el acceso ambos volúmenes únicamente a la máquina server1, cyra dirección IP es: 10.0.100.12.

```
# gluster volume set gv0 auth.allow 10.0.100.12
# gluster volume set gv1 auth.allow 10.0.100.12
```

6. [cluster2] Montaje y utilización de los volúmenes GlusterFS.

Los dos volúmenes *glusterFS* creados los montaremos en la máquina `cluster2`. Para ello, primero instalaremos los paquetes necesarios:

```
# yum –y install centos-release-gluster
# yum install –y glusterfs-client
# glusterfs --version
```

Para que no haya problemas las versiones de *gluster* instaladas en los servidores y en el/los clientes han de ser iguales.

Ahora, crearemos sendos puntos de montaje:

```
# mkdir -p /mnt/gfs-r
# mkdir –p /mnt/gfs-d
```

Y montamos los dos volúmenes, replicado y distribuido, respectivamente:

```
# mount –t glusterfs cluster3:/gv0 /mnt/gfs-r
# mount –t glusterfs cluster3:/gv1 /mnt/gfs-d
```

Ahora podemos crear algunos archivos para observar el funcionamiento de GlusterFS.

La siguiente orden crea 100 copias de un archivo (por ejemplo `/etc/hosts`) en el volumen replicado:

```
# for i in `seq 1 100`; do cp /etc/hosts /mnt/gfs-r/replica$i; done
```

Si obtenemos un listado del contenido del *brick* correspondiente en cada una de las máquinas, observaremos que todas tienen los 100 archivos creados, puesto que el volumen funciona como replicado. Esto es, en las máquinas `cluster3` y `cluster4`, el listado mostrado debe ser el mismo:

```
# ls -la /export/brick1/data
```

La siguiente órden crea 100 copias de un archivo (por ejemplo `/etc/hosts`) en el volumen distribuido:

```
# for i in `seq 1 100`; do cp /etc/hosts /mnt/gfs-d/distr$i; done
```

Si obtenemos un listado del contenido del *brick* correspondiente en cada una de las máquinas, observaremos que los archivos creados se han repartido entre los dos *bricks* que componen el volumen. Esto es, en las máquinas `cluster3` y `cluster4`, el listado mostrado por el siguiente comando es distinto:

```
# ls -la /export/brick2/data
```

De hecho, la unión de los archivos mostrados en cada una de las máquinas deben ser los 100 archivos que hemos creado.

Para finalizar, si deseáramos consolidar la configuración, deberíamos crear las entradas correspondientes en el archivo `/etc/fstab`.

Vamos a simular la caída de uno de los servidores de almacenamiento, por ejemplo, `cluster3`. Apagamos la máquina virtual `cluster3` y desde `cluster2` lanzamos las ordenes:

```
# ls -la /mnt/gfs-r
```

```
# ls -la /mnt/gfs-d
```

Muestre la salida obtenida en cada caso y explique a que es debido.