

Despliegue de un servicio de archivos distribuido sobre un clúster virtualizado.

Se va a construir un clúster virtualizado sobre OpenNebula de 4 MV para crear un sistema de archivos distribuido que tolere errores, y probar su rendimiento.

1. Utilizando Debian11, crear 4 MV: 1 frontend con dos IPs (pública y privada), y 3 MV (nodos) con IP privadas solo dado que estas MV se tendrán que conectar y configurar desde el frontend.

The screenshot shows the 'Virtual Machines' management interface. At the top, there's a green '+ VM' button, a search bar labeled 'Search VMs', and a dropdown menu set to 'ALL'. Below, four VM cards are displayed:

- Frontend**: x1 - 768MB - Debian 11, IP 192.168.0.11, 84.88.58.69. Created 5 Mar.
- nodo1**: x1 - 768MB - Debian 11, IP 192.168.0.14. Created 4 Mar.
- nodo2**: x1 - 768MB - Debian 11, IP 192.168.0.15. Created 2 Mar.
- nodo3**: x1 - 768MB - Debian 11, IP 192.168.0.16. Created 2 Mar.

Each card includes a status icon (a small black square), a user icon labeled 'ajustr', and a date.

Se han creado 1 Server con ip pública y otra privada; además 3 nodos sólo con clave privada.

Accediendo al Server desde local

```
→ ~ ssh -i .ssh/id_rsa root@84.88.58.69 -p 55000
Linux localhost.localdomain 5.10.0-28-amd64 #1 SMP Debian 5.10.209-2
(2024-01-31) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar 24 08:47:32 2024 from 85.87.66.72
root@localhost:~#
```

2. Sobre las 3 MV (tipo nodo) añadir un disco adicional del tipo volátil y en el Bus Virtio (advanced options) de 2GB, crear una partición, un filesystem ext4 y montarlo de forma permanente (/etc/fstab) en cada MV en /export/brick.

Attach new disk

Attach new disk

☐ Image

☒ Volatile disk

Size

2

GB

Disk type

FS

Filesystem format

raw

^ Advanced options

Target device

sdc

Image mapping driver

BUS

Virtio

Read-only

Partición creada.







Capacity

Storage

Network

Snapshots

Actions

ID	Target	Image / Size-Format	Size	Persistent	Actions	Attach disk
0	vda	Debian 11	1.9GB/12GB	NO	   	
1	hda	Context	1MB/-	NO		
2	vdb	2GB - raw	0MB/2GB	NO	 	

Repetimos le proceso en los tres nodos.

```
# accedo a nodo 1 desde Server
root@localhost:~# ssh -i /root/.ssh/id_rsa root@192.168.0.14 -p 55000
Linux nodo1 5.10.0-28-amd64 #1 SMP Debian 5.10.209-2 (2024-01-31)
x86_64

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar 22 18:16:44 2024 from 192.168.0.11

root@nodo1:~#
```

Verificar que el disco volátil adicional ha sido reconocido por el sistema como **/dev/vdb**

```
root@nodo1:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1   364K  0 rom
vda         254:0    0    12G  0 disk
```

```
| -vda1 254:1 0 11.9G 0 part /  
| -vda14 254:14 0 3M 0 part  
`-vda15 254:15 0 124M 0 part /boot/efi  
vdb 254:16 0 2G 0 disk  
root@nodo1:~#
```

`lsblk` muestra que `vdb` es el nombre del nuevo disco de 2GB que se ha añadido, y que aún no tiene ninguna partición ni punto de montaje asignado. Ahora se procede con la creación de la partición, el sistema de archivos y el montaje.

```
# Crear una nueva partición  
root@nodo1:~# sudo fdisk /dev/vdb  
  
Welcome to fdisk (util-linux 2.36.1).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.  
  
Device does not contain a recognized partition table.  
Created a new DOS disklabel with disk identifier 0x52d2fae9.  
  
Command (m for help): n  
Partition type  
p primary (0 primary, 0 extended, 4 free)  
e extended (container for logical partitions)  
Select (default p): p  
Partition number (1-4, default 1): 1  
First sector (2048-4194303, default 2048):  
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default  
4194303):  
  
Created a new partition 1 of type 'Linux' and of size 2 GiB.  
  
Command (m for help): w  
The partition table has been altered.  
Calling ioctl() to re-read partition table.  
Syncing disks.  
  
root@nodo1:~#
```

Formatear la nueva partición como ext4

Este comando creará un sistema de archivos ext4 en la nueva partición (que se asume que es vdb1).

```
root@nodo1:~# sudo mkfs.ext4 /dev/vdb1  
mke2fs 1.46.2 (28-Feb-2021)  
Creating filesystem with 524032 4k blocks and 131072 inodes  
Filesystem UUID: 78421551-c083-46a6-97e3-e94a2d96d888  
Superblock backups stored on blocks:  
32768, 98304, 163840, 229376, 294912
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
root@nodo1:~#
```

Crear el punto de montaje

Esto añade la nueva partición al archivo de sistema de ficheros de tabla (/etc/fstab) para que se monte automáticamente en el arranque.

```
# Crear el punto de montaje
root@nodo1:~# sudo mkdir -p /export/brick

# añadir carpeta para el montaje automático
root@nodo1:~# echo '/dev/vdb1 /export/brick ext4 defaults 0 2' | sudo tee
-a /etc/fstab

/dev/vdb1 /export/brick ext4 defaults 0 2

# Montar la partición según las entradas en /etc/fstab
root@nodo1:~# sudo mount -a

# verificamos
root@nodo1:~# df -h /export/brick
    Filesystem      Size  Used Avail Use% Mounted on
    /dev/vdb1        2.0G   24K   1.9G   1% /export/brick
```

La salida del comando `df -h /export/brick` confirma que la partición `/dev/vdb1` ha sido montada correctamente en el directorio `/export/brick`, y tienes aproximadamente 1.9 GB de espacio disponible para usar, lo que indica que el sistema de archivos ext4 ha sido creado y montado exitosamente.

Repetimos para cada nodo los mismo pasos

nodo2

```
root@localhost:~# ssh -i /root/.ssh/id_rsa root@192.168.0.15 -p 55000
Linux nodo2 5.10.0-28-amd64 #1 SMP Debian 5.10.209-2 (2024-01-31)
x86_64

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar 22 18:16:59 2024 from 192.168.0.11

root@nodo2:~# lsblk
    NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
    sr0          11:0    1   364K  0 rom
```

```
vda      254:0      0   12G   0 disk
|-vda1   254:1      0  11.9G   0 part /
|-vda14  254:14     0    3M   0 part
`-vda15  254:15     0   124M   0 part /boot/efi
vdb      254:16     0    2G   0 disk
```

```
root@nodo2:~# sudo fdisk /dev/vdb
```

```
Welcome to fdisk (util-linux 2.36.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xfd46121f.
```

```
Command (m for help): n
Partition type
p   primary (0 primary, 0 extended, 4 free)
e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default
4194303):
```

```
Created a new partition 1 of type 'Linux' and of size 2 GiB.
```

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

```
root@nodo2:~# sudo mkfs.ext4 /dev/vdb1
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 524032 4k blocks and 131072 inodes
Filesystem UUID: 8dd4d27a-c3b9-4bae-80f2-58cd50c86d28
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
root@nodo2:~# sudo mkdir -p /export/brick
root@nodo2:~# echo '/dev/vdb1 /export/brick ext4 defaults 0 2' | sudo tee
-a /etc/fstab
    /dev/vdb1 /export/brick ext4 defaults 0 2
root@nodo2:~# sudo mount -a
root@nodo2:~# df -h /export/brick
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/vdb1	2.0G	24K	1.9G	1%	/export/brick

nodo3

```
root@localhost:~# ssh -i /root/.ssh/id_rsa root@192.168.0.16 -p 55000
Linux nodo3 5.10.0-28-amd64 #1 SMP Debian 5.10.209-2 (2024-01-31)
x86_64
```

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Fri Mar 22 18:17:08 2024 from 192.168.0.11

```
root@nodo3:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1   364K  0 rom
vda          254:0    0    12G  0 disk
|-vda1       254:1    0  11.9G  0 part /
|-vda14      254:14   0     3M  0 part
`-vda15      254:15   0   124M  0 part /boot/efi
vdb          254:16   0     2G  0 disk
```

```
root@nodo3:~# sudo fdisk /dev/vdb
```

Welcome to fdisk (util-linux 2.36.1).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.

Created a new DOS disklabel with disk identifier 0x62958d32.

Command (m for help): n

Partition type

p primary (0 primary, 0 extended, 4 free)

e extended (container for logical partitions)

Select (default p): p

Partition number (1-4, default 1): 1

First sector (2048-4194303, default 2048):

Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-4194303, default 4194303):

Created a new partition 1 of type 'Linux' and of size 2 GiB.

Command (m for help): w

The partition table has been altered.

Calling ioctl() to re-read partition table.

Syncing disks.

```
root@nodo3:~# sudo mkfs.ext4 /dev/vdb1
```

```
mke2fs 1.46.2 (28-Feb-2021)
Creating filesystem with 524032 4k blocks and 131072 inodes
Filesystem UUID: 9359147b-0417-4b77-994e-ac9313114918
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
root@nodo3:~# sudo mkdir -p /export/brick
root@nodo3:~# echo '/dev/vdb1 /export/brick ext4 defaults 0 2' | sudo tee
-a /etc/fstab
    /dev/vdb1 /export/brick ext4 defaults 0 2
root@nodo3:~# sudo mount -a
root@nodo3:~# df -h /export/brick
Filesystem      Size  Used Avail Use% Mounted on
/dev/vdb1       2.0G   24K  1.9G   1% /export/brick
```

3. Configurar Frontend para que haga un ip_forward y NAT para que Nodo1-3 tengan conexión a Internet a través de él.

Habilitar IP forwarding

```
root@localhost:~# nano /etc/sysctl.conf

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# aplico cambios
root@localhost:~# sysctl -p
```

Configurar NAT con iptables

Ya se configuró en la primera parte "Virtual proxy balancer"

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
root@localhost:~# sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 366 packets, 27830 bytes)
 pkts bytes target    prot opt in     out     source
destination

Chain INPUT (policy ACCEPT 366 packets, 27830 bytes)
 pkts bytes target    prot opt in     out     source
destination
```

```
Chain OUTPUT (policy ACCEPT 147 packets, 11056 bytes)
pkts bytes target      prot opt in      out      source
destination

Chain POSTROUTING (policy ACCEPT 5 packets, 300 bytes)
pkts bytes target      prot opt in      out      source
destination
142 10756 MASQUERADE  all  --  *      eth1     0.0.0.0/0
0.0.0.0/0
```

La salida muestra que la regla de NAT utilizando MASQUERADE está de hecho aplicada en la interfaz **eth1**, no en eth0 como se mencionó originalmente. Esto significa que cualquier tráfico que salga de la interfaz eth1 será enmascarado, lo que permite a las máquinas detrás de esta interfaz acceder a Internet o a otra red externa.

```
root@localhost:~# ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=119 time=10.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=9.94 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=119 time=9.96 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=119 time=9.92 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 9.917/9.964/10.042/0.047 ms

root@localhost:~# traceroute google.com
traceroute to google.com (142.250.201.78), 30 hops max, 60 byte
packets
 1  84.88.58.65 (84.88.58.65)  0.381 ms  0.635 ms  0.252 ms
 2  * * *
 3  anella-uoc-lab.cesca.cat (84.88.19.165)  1.507 ms  1.571 ms  1.377
ms
 4  google.02.catnix.net (193.242.98.156)  9.779 ms  9.562 ms  9.360 ms
 5  142.251.53.181 (142.251.53.181)  9.629 ms  9.676 ms  9.473 ms
 6  74.125.37.87 (74.125.37.87)  9.636 ms  142.250.232.7 (142.250.232.7)
9.913 ms 74.125.37.87 (74.125.37.87)  9.706 ms
 7  mad07s25-in-f14.1e100.net (142.250.201.78)  9.824 ms  9.639 ms
9.726 ms
```

Pasos siguientes:

- Verificar configuración de los nodos: nodos configurados con el frontend como su gateway por defecto y que puedan realizar ping y traceroute a direcciones en Internet para confirmar su conectividad.
- Mantenimiento de la seguridad: las reglas de firewall y NAT no expongan innecesariamente servicios o puertos internos a Internet, especialmente si eth1 está conectada directamente a Internet.

nodo 1


```
root@localhost:~# ssh -i /root/.ssh/id_rsa root@192.168.0.14 -p 55000

root@nodo1:~# sudo ip route add default via 192.168.0.11

root@nodo1:~# ip route show
default via 192.168.0.11 dev eth0
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.14

root@nodo1:~# ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=10.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=10.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=10.7 ms

root@nodo1:~# traceroute google.com
traceroute to google.com (216.58.215.142), 30 hops max, 60 byte
packets
 1  192.168.0.11 (192.168.0.11)  0.712 ms  0.539 ms  0.587 ms
 2  84.88.58.65 (84.88.58.65)  1.126 ms  0.829 ms  1.693 ms
 3  * * *
 4  anella-uoc-lab.cesca.cat (84.88.19.165)  2.727 ms  2.569 ms  2.881
ms
 5  google.02.catnix.net (193.242.98.156)  10.950 ms  10.803 ms  10.655
ms
 6  192.178.110.87 (192.178.110.87)  11.685 ms  11.242 ms  11.125 ms
 7  142.250.239.25 (142.250.239.25)  11.180 ms  11.031 ms  11.032 ms
 8  mad41s04-in-f14.1e100.net (216.58.215.142)  10.287 ms  10.486 ms
10.315 ms
```

nodo 2

```
root@localhost:~# ssh -i /root/.ssh/id_rsa root@192.168.0.15 -p 55000
Linux nodo2 5.10.0-28-amd64 #1 SMP Debian 5.10.209-2 (2024-01-31)
x86_64

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Apr 13 16:10:40 2024 from 192.168.0.11

root@nodo2:~# ip route show
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.15

root@nodo2:~# sudo ip route add default via 192.168.0.11
root@nodo2:~# ip route show
default via 192.168.0.11 dev eth0
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.15

root@nodo2:~# ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=10.6 ms
```

```

64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=10.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=10.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=10.6 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 10.583/10.625/10.650/0.027 ms

```

```

root@nodo2:~# traceroute google.com
traceroute to google.com (142.250.200.142), 30 hops max, 60 byte
packets
 1  192.168.0.11 (192.168.0.11)  0.595 ms  0.657 ms  0.596 ms
 2  84.88.58.65 (84.88.58.65)  1.358 ms  1.196 ms  1.114 ms
 3  * * *
 4  anella-uoc-lab.cesca.cat (84.88.19.165)  2.887 ms  2.723 ms  2.423
ms
 5  google.02.catnix.net (193.242.98.156)  10.579 ms  10.427 ms  10.276
ms
 6  192.178.110.73 (192.178.110.73)  10.373 ms  192.178.110.87
(192.178.110.87)  11.530 ms  11.538 ms
 7  142.251.51.143 (142.251.51.143)  11.060 ms  11.025 ms  11.096 ms
 8  mad41s14-in-f14.1e100.net (142.250.200.142)  10.917 ms  10.890 ms
10.355 ms

```

nodo 3

```

root@localhost:~# ssh -i /root/.ssh/id_rsa root@192.168.0.16 -p 55000
Linux nodo3 5.10.0-28-amd64 #1 SMP Debian 5.10.209-2 (2024-01-31)
x86_64

```

```

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Apr 13 16:14:58 2024 from 192.168.0.11

```

```

root@nodo3:~# ip route show
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.16
root@nodo3:~# sudo ip route add default via 192.168.0.11
root@nodo3:~# ip route show
default via 192.168.0.11 dev eth0
192.168.0.0/24 dev eth0 proto kernel scope link src 192.168.0.16
root@nodo3:~# ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=10.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=10.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=10.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=10.7 ms
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 10.662/10.694/10.717/0.020 ms

```

```

root@nodo3:~# traceroute google.com
traceroute to google.com (142.250.200.142), 30 hops max, 60 byte
packets

```

```

1 192.168.0.11 (192.168.0.11) 0.787 ms 0.608 ms 0.692 ms
2 84.88.58.65 (84.88.58.65) 1.034 ms 0.869 ms 1.470 ms
3 * * *
4 anella-uoc-lab.cesca.cat (84.88.19.165) 2.788 ms 2.607 ms 2.430
ms
5 google.02.catnix.net (193.242.98.156) 10.489 ms 10.327 ms 10.326
ms
6 192.178.110.73 (192.178.110.73) 10.718 ms 192.178.110.87
(192.178.110.87) 11.639 ms 11.810 ms
7 142.251.51.141 (142.251.51.141) 10.829 ms 10.669 ms
142.251.51.143 (142.251.51.143) 10.786 ms
8 mad41s14-in-f14.1e100.net (142.250.200.142) 10.985 ms 10.356 ms
11.184 ms

```

4. Actualizar Debian11 e instalar GlusterFS Server sobre las 3 MV (tipo nodo) para que se transformen en servidores del sistema de archivo distribuidos.

Conecto a nodo 1, 2 y 3 para actualizar cada sistema

```

# actualizar sistema de paquetes
root@nodo1:~# "apt update && apt upgrade -y && apt autoremove -y"

# actualizar sistema de paquetes
root@nodo2:~# "apt update && apt upgrade -y && apt autoremove -y"

# actualizar sistema de paquetes
root@nodo3:~# "apt update && apt upgrade -y && apt autoremove -y"

```

Instalar GlusterFS en Cada Nodo

```

root@nodo1:~# apt install glusterfs-server -y && systemctl start
glusterfs-server && systemctl enable glusterfs-server

root@nodo2:~# apt install glusterfs-server -y && systemctl start
glusterfs-server && systemctl enable glusterfs-server

root@nodo3:~# apt install glusterfs-server -y && systemctl start
glusterfs-server && systemctl enable glusterfs-server

```

Configurar GlusterFS

Conectar los nodos entre sí: establecer la conectividad entre los nodos desde uno de los nodos

En cada nodo

```
root@nodo1:~# sudo systemctl start glusterd
root@nodo1:~# sudo systemctl enable glusterd
Created symlink /etc/systemd/system/multi-
user.target.wants/glusterd.service → /lib/systemd/system/glusterd.service.
root@nodo1:~# sudo systemctl status glusterd
● glusterd.service - GlusterFS, a clustered file-system server
   Loaded: loaded (/lib/systemd/system/glusterd.service; enabled;
 vendor preset: enabled)
   Active: active (running) since Sat 2024-04-13 17:05:31 UTC; 14s
 ago
     Docs: man:glusterd(8)
    Main PID: 7226 (glusterd)
      Tasks: 9 (limit: 808)
     Memory: 6.6M
        CPU: 1.729s
    CGroup: /system.slice/glusterd.service
            └─7226 /usr/sbin/glusterd -p /var/run/glusterd.pid --log-
level INFO

Apr 13 17:05:29 nodo1 systemd[1]: Starting GlusterFS, a clustered
file-system server...
Apr 13 17:05:31 nodo1 systemd[1]: Started GlusterFS, a clustered file-
system server.
root@nodo1:~#
```

```
root@nodo2:~# sudo systemctl start glusterd
root@nodo2:~# sudo systemctl enable glusterd
Created symlink /etc/systemd/system/multi-
user.target.wants/glusterd.service → /lib/systemd/system/glusterd.service.
root@nodo2:~# sudo systemctl status glusterd
● glusterd.service - GlusterFS, a clustered file-system server
   Loaded: loaded (/lib/systemd/system/glusterd.service; enabled;
 vendor preset: enabled)
   Active: active (running) since Sat 2024-04-13 17:07:07 UTC; 9s ago
     Docs: man:glusterd(8)
    Main PID: 7262 (glusterd)
      Tasks: 9 (limit: 808)
     Memory: 10.4M
        CPU: 1.865s
    CGroup: /system.slice/glusterd.service
            └─7262 /usr/sbin/glusterd -p /var/run/glusterd.pid --log-
level INFO

Apr 13 17:07:04 nodo2 systemd[1]: Starting GlusterFS, a clustered
file-system server...
Apr 13 17:07:07 nodo2 systemd[1]: Started GlusterFS, a clustered file-
system server.
```

```
root@nodo3:~# sudo systemctl start glusterd
root@nodo3:~# sudo systemctl enable glusterd
Created symlink /etc/systemd/system/multi-
```

```
user.target.wants/glusterd.service → /lib/systemd/system/glusterd.service.
```

```
root@nodo3:~# sudo systemctl status glusterd
```

```
● glusterd.service – GlusterFS, a clustered file-system server
   Loaded: loaded (/lib/systemd/system/glusterd.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Sat 2024-04-13 17:08:09 UTC; 9s ago
   Docs: man:glusterd(8)
   Main PID: 7212 (glusterd)
   Tasks: 9 (limit: 808)
   Memory: 10.6M
   CPU: 1.952s
   CGroup: /system.slice/glusterd.service
           └─7212 /usr/sbin/glusterd -p /var/run/glusterd.pid --log-
level INFO
```

```
Apr 13 17:08:07 nodo3 systemd[1]: Starting GlusterFS, a clustered
file-system server...
```

```
Apr 13 17:08:09 nodo3 systemd[1]: Started GlusterFS, a clustered file-
system server.
```

para cada nodo

```
root@nodo1:~# sudo nano /etc/hosts
```

```
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters

192.168.0.11   frontend
192.168.0.14   nodo1
192.168.0.15   nodo2
192.168.0.16   nodo3
```

Verificar la Conectividad

```
root@nodo1:~# ping -c 4 nodo2
```

```
PING nodo2 (192.168.0.15) 56(84) bytes of data.
64 bytes from nodo2 (192.168.0.15): icmp_seq=1 ttl=64 time=1.72 ms
64 bytes from nodo2 (192.168.0.15): icmp_seq=2 ttl=64 time=0.795 ms
64 bytes from nodo2 (192.168.0.15): icmp_seq=3 ttl=64 time=0.789 ms
64 bytes from nodo2 (192.168.0.15): icmp_seq=4 ttl=64 time=0.753 ms
--- nodo2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.753/1.013/1.718/0.406 ms
```

```
root@nodo1:~# ping -c 4 nodo3
```

```
PING nodo3 (192.168.0.16) 56(84) bytes of data.
```

```
64 bytes from nodo3 (192.168.0.16): icmp_seq=1 ttl=64 time=1.86 ms
64 bytes from nodo3 (192.168.0.16): icmp_seq=2 ttl=64 time=0.755 ms
64 bytes from nodo3 (192.168.0.16): icmp_seq=3 ttl=64 time=0.777 ms
64 bytes from nodo3 (192.168.0.16): icmp_seq=4 ttl=64 time=0.798 ms
--- nodo3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.755/1.047/1.860/0.469 ms
```

Reconectar los Nodos en GlusterFS

```
root@nodo1:~# gluster peer probe nodo2
peer probe: success

root@nodo1:~# gluster peer probe nodo3
peer probe: success

root@nodo1:~# gluster peer status
Number of Peers: 2

Hostname: nodo2
Uuid: d9d1032b-092e-4f7e-a94e-9a3bbe8e6ee3
State: Peer in Cluster (Connected)

Hostname: nodo3
Uuid: 87e3e964-d8e6-46c9-8333-7b18a756ee60
State: Peer in Cluster (Connected)
```

5. Configurar las 3 MV (tipo nodo) como servidores del sistema de archivos distribuido y crear un volumen sobre todos los discos añadidos con un factor de réplica 2 y sin NFS. Instalar sobre la MV con IP pública (frontend) GlusterFS Client, montar el volumen creado y realizar pruebas funcionales verificando que los archivos copiados se encuentran tantos como réplicas en los diferentes servidores de archivos.

En los Nodos de GlusterFS (nodo1, nodo2, nodo3):

```
# Crear Directorios de Bricks:
root@nodo1:~# mkdir -p /export/brick1/gv0
root@nodo2:~# mkdir -p /export/brick1/gv0
root@nodo3:~# mkdir -p /export/brick1/gv0
```

Crear un Volumen con Réplica 2

Aquí, gv0 es el nombre del volumen y `/export/brick1/gv0` es la ruta al 'brick' en cada nodo. La palabra clave `force` se usa para forzar la creación del volumen si Gluster detecta algún problema potencial con la configuración.

```
# crear un volumen replicado en nodo1
root@nodo1:~# gluster volume create gv0 replica 2 nodo1:/export/brick1/gv0
nodo2:/export/brick1/gv0 force

volume create: gv0: success: please start the volume to access data
```

Iniciar el Volumen

```
root@nodo1:~# gluster volume start gv0
volume start: gv0: success

# verifica
root@nodo1:~# gluster volume info

Volume Name: gv0
Type: Replicate
Volume ID: 3927cb9f-86f0-447b-9055-da8ac7369fe6
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: nodo1:/export/brick1/gv0
Brick2: nodo2:/export/brick1/gv0
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

En el Frontend (GlusterFS Client)

```
# Instalar GlusterFS Client
apt update
apt install glusterfs-client
```

```
root@localhost:~# gluster peer probe 192.168.0.16
peer probe: success

root@localhost:~# gluster peer status
Number of Peers: 2

Hostname: 192.168.0.15
Uuid: 757b9098-934d-41cc-8f78-27bd479b9a3f
State: Peer in Cluster (Connected)
```

```
Hostname: 192.168.0.16
Uuid: 2273c67e-5a95-4c8b-aa09-bed21cb537ab
State: Peer in Cluster (Connected)
```

los nodos nodo2 y nodo3 están conectados y forman parte del clúster de GlusterFS.

Creación del Volumen en GlusterFS

```
# Crear Directorios de Bricks:
root@nodo2:~# "sudo mkdir -p /export/brick1/gv0"
root@nodo3:~# "sudo mkdir -p /export/brick1/gv0"

root@localhost:~# gluster volume create gv0 replica 2
192.168.0.15:/export/brick1/gv0 192.168.0.16:/export/brick1/gv0
Replica 2 volumes are prone to split-brain. Use Arbiter or Replica 3
to avoid this. See:
http://docs.gluster.org/en/latest/Administrator%20Guide/Split%20brain%20and%20ways%20to%20deal%20with%20it/.
Do you still want to continue?
(y/n) y
volume create: gv0: failed: Staging failed on 192.168.0.15. Error: The
brick 192.168.0.15:/export/brick1/gv0 is being created in the root
partition. It is recommended that you don't use the system's root
partition for storage backend. Or use 'force' at the end of the command if
you want to override this behavior.
Staging failed on 192.168.0.16. Error: The brick
192.168.0.16:/export/brick1/gv0 is being created in the root partition. It
is recommended that you don't use the system's root partition for storage
backend. Or use 'force' at the end of the command if you want to override
this behavior.
root@localhost:~#
```

Verificar configuración de los servidores GlusterFS en los nodos

Primero, necesitas verificar si los nodos están configurados correctamente como servidores de GlusterFS y si ya están funcionando como un cluster. En cualquiera de los nodos.

```
root@nodo1:~# gluster peer status
Number of Peers: 2

Hostname: nodo2
Uuid: d9d1032b-092e-4f7e-a94e-9a3bbe8e6ee3
State: Peer Rejected (Connected)

Hostname: nodo3
Uuid: 87e3e964-d8e6-46c9-8333-7b18a756ee60
State: Peer Rejected (Connected)
```


El estado **Peer Rejected (Connected)** indica que los nodos están conectados, pero hay un conflicto en la configuración o una discrepancia en los volúmenes entre los nodos que debe ser resuelto.

Verificar la creación del volumen de GlusterFS

Confirmar que el volumen de GlusterFS ya ha sido creado y configurado con el factor de réplica correcto.

```
root@nodo1:~# gluster volume info
Volume Name: gv0
Type: Replicate
Volume ID: 3927cb9f-86f0-447b-9055-da8ac7369fe6
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: nodo1:/export/brick1/gv0
Brick2: nodo2:/export/brick1/gv0
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

Comprobar la instalación de GlusterFS Client en el frontend

```
root@localhost:~# glusterfs --version
glusterfs 9.2
Repository revision: git://git.gluster.org/glusterfs.git
Copyright (c) 2006-2016 Red Hat, Inc. <https://www.gluster.org/>
GlusterFS comes with ABSOLUTELY NO WARRANTY.
It is licensed to you under your choice of the GNU Lesser
General Public License, version 3 or any later version (LGPLv3
or later), or the GNU General Public License, version 2 (GPLv2),
in all cases as published by the Free Software Foundation.
```

Verificar el montaje del volumen de GlusterFS en el frontend*

Es importante confirmar si el volumen de GlusterFS ya está montado en el sistema frontend.

```
root@localhost:~# mount | grep glusterfs
root@localhost:~#
```

```
root@nodo1:~# gluster peer status
Number of Peers: 2
```

```
Hostname: nodo2
Uuid: d9d1032b-092e-4f7e-a94e-9a3bbe8e6ee3
State: Peer Rejected (Connected)
```

```
Hostname: nodo3
Uuid: 87e3e964-d8e6-46c9-8333-7b18a756ee60
State: Peer Rejected (Connected)
```

Revisar los registros de GlusterFS

sería útil revisar los registros de GlusterFS en nodo1, nodo2, y nodo3 para ver si hay mensajes de error

```
cat /var/log/glusterfs/glusterd.log
```

Problemas identificados:

Errores de Resolución de DNS: Hay errores recurrentes relacionados con la incapacidad para resolver los nombres de host (Name or service not known). Esto sugiere un problema de resolución de nombres en la configuración de red o DNS.

Archivos Faltantes: Varios errores indican la falta de archivos en el directorio `/var/lib/glusterd/`, lo cual podría estar interfiriendo con la correcta gestión del cluster.

Errores de UUID y Quórum: Hay referencias a problemas con los UUID de los nodos y el quórum del servidor, lo que indica que la configuración del cluster puede estar incompleta o corrupta.

Verificar y configurar DNS o `/etc/hosts`:

```
root@localhost:~# cat /etc/hosts
127.0.0.1      localhost localhost.localdomain
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
84.88.58.69    proxy.netum.org

# añadido esto
192.168.0.14  nodo1
192.168.0.15  nodo2
192.168.0.16  nodo3
192.168.0.11  frontend

root@localhost:~# nc -zv nodo2 24007
Connection to nodo2 (192.168.0.15) 24007 port [tcp/*] succeeded!
root@localhost:~# nc -zv nodo3 24007
Connection to nodo3 (192.168.0.16) 24007 port [tcp/*] succeeded!
root@localhost:~#
```

Crear un Volumen en GlusterFS replica 2

```
# creo volume factor replica 2
root@localhost:~# gluster volume create gv0 replica 2 transport tcp \
> nodo1:/data/brick1/gv0 \
> nodo2:/data/brick1/gv0 force
volume create: gv0: success: please start the volume to access data

root@localhost:~# gluster volume start gv0
volume start: gv0: success

root@localhost:~# gluster volume info gv0
Volume Name: gv0
Type: Replicate
Volume ID: a49d5e0b-64b2-4868-a478-89d9d5535845
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: nodo1:/data/brick1/gv0
Brick2: nodo2:/data/brick1/gv0
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off
```

El volumen está configurado para replicar los datos entre nodo1 y nodo2, lo cual mejora la redundancia y la disponibilidad de tus datos en caso de que uno de los nodos falle.

```
root@localhost:~# gluster volume status gv0
Status of volume: gv0
Gluster process                                TCP Port  RDMA Port
Online  Pid
-----
-----
    Brick nodo1:/data/brick1/gv0                49152      0          Y
11744
    Brick nodo2:/data/brick1/gv0                49153      0          Y
11596
    Self-heal Daemon on localhost                N/A       N/A        Y
17649
    Self-heal Daemon on 192.168.0.16             N/A       N/A        Y
8748
    Self-heal Daemon on nodo1                    N/A       N/A        Y
11761
    Self-heal Daemon on 192.168.0.15             N/A       N/A        Y
11613
```

Task Status of Volume gv0

There are no active volume tasks

Verificación final que estén todos los pasos

```
# Verificar Instalación del Cliente GlusterFS en el Frontend
root@localhost:~# glusterfs --version
glusterfs 9.2
Repository revision: git://git.gluster.org/glusterfs.git
Copyright (c) 2006-2016 Red Hat, Inc. <https://www.gluster.org/>
GlusterFS comes with ABSOLUTELY NO WARRANTY.
It is licensed to you under your choice of the GNU Lesser
General Public License, version 3 or any later version (LGPLv3
or later), or the GNU General Public License, version 2 (GPLv2),
in all cases as published by the Free Software Foundation.

# Verificar Montaje del Volumen
root@localhost:~# mount | grep gv0
nodo1:/gv0 on /mnt/gv0 type fuse.glusterfs
(rw,relatime,user_id=0,group_id=0,default_permissions,allow_other,max_read
=131072)

# Realizar Pruebas Funcionales
root@nodo1:~# cat /mnt/gv0/testfile.txt
cat: /mnt/gv0/testfile.txt: Transport endpoint is not connected
root@nodo1:~# umount /mnt/gv0
root@nodo1:~# mount -t glusterfs nodo1:/gv0 /mnt/gv0
root@nodo1:~# cat /mnt/gv0/testfile.txt
Hello, Gluster!

root@nodo2:~# umount /mnt/gv0
root@nodo2:~# mount -t glusterfs nodo1:/gv0 /mnt/gv0
root@nodo2:~# cat /mnt/gv0/testfile.txt
Hello, Gluster!
```

6. Sobre la MV frontend realizar pruebas de rendimiento de lectura y escritura utilizando algunas de las herramientas sugeridas por los desarrolladores para hacer performance testing (<https://glusterdocs.readthedocs.io/en/latest/Administrator%20Guide/Performance%20Testing/>) del sistema de archivo distribuido como por ejemplo iozones3 o fio.

```
root@localhost:~# sudo apt update
root@localhost:~# sudo apt install fio
root@localhost:~# sudo find / -name fio
/usr/bin/fio
```

```
/usr/share/bash-completion/completions/fio
/usr/share/doc/fio
/usr/share/doc-base/fio
/usr/share/fio
find: '/mnt/gv0': Transport endpoint is not connected
/etc/init.d/fio
root@localhost:~# export PATH=$PATH:/usr/bin/fio
```

Instalo en cada nodo fio

```
# instalo
root@nodo1:~# sudo apt update
root@nodo1:~# sudo apt install fio -y

root@nodo2:~# sudo apt update
root@nodo2:~# sudo apt install fio -y
```

```
# activo fio
root@nodo1:~# /usr/bin/fio --server --daemonize=/var/run/fio-svr.pid
root@nodo2:~# /usr/bin/fio --server --daemonize=/var/run/fio-svr.pid
```

Ejecutar pruebas de rendimiento

Paso 1: Crear un archivo de configuración para fio

```
root@localhost:~# nano test.fio
root@localhost:~# cat test.fio
[global]
bs=4k
iodepth=16
direct=1
ioengine=libaio
time_based
runtime=60
size=1G

[job1]
rw=write
filename=/mnt/gv0/testfile1

[job2]
rw=read
filename=/mnt/gv0/testfile1
```

Paso 2: Ejecutar fio en modo cliente desde el frontend

Asegúrate de que fio está corriendo en modo servidor en los nodos de GlusterFS (nodo1 y nodo2).

```

root@localhost:~# fio --client=192.168.0.14 --client=192.168.0.15 test.fio
  hostname=nodo2, be=0, 64-bit, os=Linux, arch=x86-64, fio=fio-3.25,
flags=1
  <nodo2> job1: (g=0): rw=write, bs=(R) 4096B-4096B, (W) 4096B-4096B,
(T) 4096B-4096B, ioengine=libaio, iodepth=16
  <nodo2> job2: (g=0): rw=red, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T)
4096B-4096B, ioengine=libaio, iodepth=16
  <nodo2> Starting 2 processes
  <nodo2> job1: Laying out IO file (1 file / 1024MiB)
  <nodo2> (f=2): [W(1),R(1)][100.0%][r=12.1M,w=1562k][r=3012,w=390 IOPS]
[eta 00m:00s]
  job1: (groupid=0, jobs=1): err= 0: pid=12199: Sun Apr 14 15:51:28 2024
  write: IOPS=4410, BW=17.2MiB/s (18.1MB/s)(1034MiB/60006msec); 0 zone
resets
    slat (usec): min=16, max=17754, avg=134.65, stdev=502.63
    clat (usec): min=2, max=479181, avg=3490.36, stdev=16977.58
    lat (usec): min=38, max=479223, avg=3625.65, stdev=17058.89
    clat percentiles (usec):
      | 1.00th=[ 22], 5.00th=[ 22], 10.00th=[ 22], 20.00th=[
23],
      | 30.00th=[ 31], 40.00th=[ 39], 50.00th=[ 55], 60.00th=[
112],
      | 70.00th=[ 273], 80.00th=[ 783], 90.00th=[ 1500], 95.00th=[
5604],
      | 99.00th=[ 99091], 99.50th=[112722], 99.90th=[154141], 99.95th=
[181404],
      | 99.99th=[274727]
    bw ( KiB/s): min= 1400, max=27656, per=100.00%, avg=17775.39,
stdev=4959.72, samples=119
    iops       : min= 350, max= 6914, avg=4443.85, stdev=1239.93,
samples=119
    lat (usec)  : 4=0.01%, 10=0.01%, 50=47.43%, 100=10.61%, 250=11.11%
    lat (usec)  : 500=5.28%, 750=4.96%, 1000=4.74%
    lat (msec)  : 2=7.67%, 4=2.22%, 10=1.89%, 20=0.43%, 50=0.64%
    lat (msec)  : 100=2.05%, 250=0.95%, 500=0.02%
    cpu         : usr=2.87%, sys=5.49%, ctx=378503, majf=0, minf=13
    IO depths   : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=100.0%, 32=0.0%,
>=64=0.0%
    submit      : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
    complete    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.1%, 32=0.0%, 64=0.0%,
>=64=0.0%
    issued rwts: total=0,264635,0,0 short=0,0,0,0 dropped=0,0,0,0
    latency     : target=0, window=0, percentile=100.00%, depth=16
  job2: (groupid=0, jobs=1): err= 0: pid=12200: Sun Apr 14 15:51:28 2024
  read: IOPS=1387, BW=5548KiB/s (5681kB/s)(325MiB/60005msec)
    slat (usec): min=2, max=328, avg= 5.26, stdev= 8.47
    clat (usec): min=612, max=135100, avg=11522.28, stdev=9475.43
    lat (usec): min=619, max=135112, avg=11527.92, stdev=9475.68
    clat percentiles (msec):
      | 1.00th=[ 3], 5.00th=[ 4], 10.00th=[ 4], 20.00th=[

```

```

5],
    | 30.00th=[    5], 40.00th=[    6], 50.00th=[    8], 60.00th=[
11],
    | 70.00th=[   14], 80.00th=[   19], 90.00th=[   26], 95.00th=[
32],
    | 99.00th=[   41], 99.50th=[   47], 99.90th=[   65], 99.95th=[
71],
    | 99.99th=[  109]
    bw (  KiB/s): min= 1584, max=12296, per=99.17%, avg=5502.11,
stdev=2917.82, samples=119
    iops      : min=   396, max= 3074, avg=1375.52, stdev=729.44,
samples=119
    lat (usec)  : 750=0.01%, 1000=0.04%
    lat (msec)  : 2=0.43%, 4=12.35%, 10=43.67%, 20=26.20%, 50=16.97%
    lat (msec)  : 100=0.34%, 250=0.01%
    cpu         : usr=0.75%, sys=0.96%, ctx=42628, majf=0, minf=27
    IO depths   : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=100.0%, 32=0.0%,
>=64=0.0%
    submit      : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
    complete    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.1%, 32=0.0%, 64=0.0%,
>=64=0.0%
    issued rwts: total=83231,0,0,0 short=0,0,0,0 dropped=0,0,0,0
    latency     : target=0, window=0, percentile=100.00%, depth=16

Run status group 0 (all jobs):
  READ: bw=5548KiB/s (5681kB/s), 5548KiB/s-5548KiB/s (5681kB/s-
5681kB/s), io=325MiB (341MB), run=60005-60005msec
  WRITE: bw=17.2MiB/s (18.1MB/s), 17.2MiB/s-17.2MiB/s (18.1MB/s-
18.1MB/s), io=1034MiB (1084MB), run=60006-60006msec
root@localhost:~#

```

Interpretación de los resultados de fio:

En los resultados de fio, hay varias métricas clave a considerar:

IOPS (Input/Output Operations Per Second): Muestra cuántas operaciones de E/S por segundo >se pudieron realizar. En tu caso, para la escritura (write) obtuviste un IOPS de >aproximadamente 4410 y para la lectura (read) un IOPS de aproximadamente 1387.

BW (Bandwidth): Indica el ancho de banda de la operación de E/S, es decir, la cantidad de >datos que se pueden transferir en un segundo. Para la escritura, el ancho de banda fue de >aproximadamente 17.2 MiB/s y para la lectura fue de aproximadamente 5.5 MiB/s.

Latency (Latencia): El tiempo que tarda en completarse una operación de E/S. Las latencias >se presentan en percentiles, lo que proporciona una buena visión de cómo se distribuyen >las latencias a lo largo de la ejecución de la prueba.

Si necesitas realizar más pruebas o ajustar parámetros, puedes modificar el archivo test.fio para probar diferentes patrones de lectura/escritura, tamaños de bloque, o duraciones de prueba. Además, si quieres investigar cómo la configuración de tu sistema de archivos distribuido afecta el rendimiento, puedes ajustar los parámetros de GlusterFS y observar cómo cambian estas métricas.

7. Eliminar el volumen previamente creado y crear otro incrementando el número de réplicas y que se pueda montar por NFS. Montar por NFS el volumen desde frontend repitiendo las pruebas de evaluación de rendimiento con la misma herramienta utilizada en el punto 6. Para este punto se deberá instalar y configurar el paquete NFS-Ganesha (https://www.server-world.info/en/note?os=Debian_11&p=glusterfs9&f=4).

Eliminar el volumen existente

```
root@localhost:~# umount /mnt/gv0
root@localhost:~# gluster volume stop gv0
  Stopping volume will make its data inaccessible. Do you want to
  continue? (y/n) y
  volume stop: gv0: success
root@localhost:~# gluster volume delete gv0
  Deleting volume will erase all information about the volume. Do you
  want to continue? (y/n) y
  volume delete: gv0: success
```

Instalar y configurar NFS-Ganesha

```
root@localhost:~# apt update
root@localhost:~# apt install nfs-ganesha nfs-ganesha-gluster
```

Configuración de NFS-Ganesha, creo un archivo básico de configuración para NFS-Ganesha para integrarlo con GlusterFS:

Eliminación del Volumen de GlusterFS Existente

```
root@localhost:~# gluster volume stop vol_distributed
  Stopping volume will make its data inaccessible. Do you want to
  continue? (y/n) y
  volume stop: vol_distributed: success
root@localhost:~# gluster volume delete vol_distributed
  Deleting volume will erase all information about the volume. Do you
  want to continue? (y/n) y
  volume delete: vol_distributed: success
```

Creación del Nuevo Volumen con Más Réplicas

```
root@localhost:~# gluster volume create vol_replicated replica 3
nodo1:/data/vol_replicated nodo2:/data/vol_replicated
nodo3:/data/vol_replicated force
  volume create: vol_replicated: success: please start the volume to
```



```
access data
```

```
root@localhost:~# gluster volume start vol_replicated
volume start: vol_replicated: success
```

Configuración de NFS-Ganesha

```
root@localhost:~# nano /etc/ganesha/ganesha.conf
root@localhost:~# cat /etc/ganesha/ganesha.conf
EXPORT
{
    Export_Id = 77;      # ID único para la exportación NFS
    Path = "/data/vol_replicated";    # Ruta del volumen GlusterFS
    Pseudo = "/vol_replicated";    # Ruta pseudo NFSv4
    Access_Type = RW;    # Tipo de acceso
    Squash = No_root_squash;    # Configuración de squash
    SecType = "sys";    # Tipos de seguridad
    FSAL {
        Name = GLUSTER;
        Hostname = "localhost"; # Asegúrate de que esto coincida con
        # uno de los nodos de Gluster
        Volume = "vol_replicated";
    }
}
root@localhost:~#
```

```
# Reiniciar NFS-Ganesha para aplicar los cambios:
root@localhost:~# systemctl restart nfs-ganesha
```

Montaje desde Frontend y Prueba de Rendimiento

En la máquina frontend (asegurándonos de que pueda resolver y alcanzar el nodo donde está corriendo NFS-Ganesha), se monta el volumen.

```
# Montar el volumen
root@localhost:~# mount -t nfs4 -o proto=tcp,port=2049
localhost:/vol_replicated /mnt

root@localhost:~# fio --name=test --rw=randrw --bs=4k --size=500M --
numjobs=2 --time_based --runtime=30 --group_reporting
test: (g=0): rw=randrw, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T)
4096B-4096B, ioengine=psync, iodepth=1
...
fio-3.25
Starting 2 processes
test: Laying out IO file (1 file / 500MiB)
test: Laying out IO file (1 file / 500MiB)
```

```

Jobs: 2 (f=2): [m(2)][100.0%][r=1256KiB/s,w=1452KiB/s][r=314,w=363
IOPS][eta 00m:00s]
test: (groupid=0, jobs=2): err= 0: pid=31620: Mon Apr 15 17:20:54 2024
read: IOPS=295, BW=1181KiB/s (1210kB/s)(34.6MiB/30008msec)
    clat (usec): min=152, max=171654, avg=6720.56, stdev=7613.35
    lat (usec): min=152, max=171655, avg=6721.05, stdev=7613.36
    clat percentiles (usec):
      | 1.00th=[ 281], 5.00th=[ 537], 10.00th=[ 816], 20.00th=[
2376],
      | 30.00th=[ 3654], 40.00th=[ 4948], 50.00th=[ 6128], 60.00th=[
7308],
      | 70.00th=[ 8291], 80.00th=[ 9372], 90.00th=[ 10814], 95.00th=[
15008],
      | 99.00th=[ 25035], 99.50th=[ 35390], 99.90th=[128451], 99.95th=
[137364],
      | 99.99th=[170918]
    bw ( KiB/s): min= 128, max= 1888, per=99.96%, avg=1181.03,
stdev=178.54, samples=118
    iops      : min= 32, max= 472, avg=295.22, stdev=44.63,
samples=118
    write: IOPS=304, BW=1220KiB/s (1249kB/s)(35.7MiB/30008msec); 0 zone
resets
    clat (usec): min=8, max=17538, avg=39.08, stdev=223.14
    lat (usec): min=8, max=17539, avg=39.60, stdev=223.17
    clat percentiles (usec):
      | 1.00th=[ 12], 5.00th=[ 14], 10.00th=[ 18], 20.00th=[
20],
      | 30.00th=[ 22], 40.00th=[ 25], 50.00th=[ 30], 60.00th=[
42],
      | 70.00th=[ 48], 80.00th=[ 52], 90.00th=[ 57], 95.00th=[
61],
      | 99.00th=[ 85], 99.50th=[ 109], 99.90th=[ 208], 99.95th=[
265],
      | 99.99th=[17433]
    bw ( KiB/s): min= 72, max= 2272, per=99.79%, avg=1217.02,
stdev=214.53, samples=118
    iops      : min= 18, max= 568, avg=304.20, stdev=53.63,
samples=118
    lat (usec) : 10=0.11%, 20=11.65%, 50=27.21%, 100=11.54%, 250=0.49%
    lat (usec) : 500=2.02%, 750=1.48%, 1000=3.35%
    lat (msec) : 2=2.03%, 4=7.01%, 10=25.84%, 20=6.50%, 50=0.56%
    lat (msec) : 100=0.13%, 250=0.08%
    cpu        : usr=0.33%, sys=1.70%, ctx=9013, majf=0, minf=32
    IO depths  : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%,
>=64=0.0%
    submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
    complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
    issued rwts: total=8863,9149,0,0 short=0,0,0,0 dropped=0,0,0,0
    latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
READ: bw=1181KiB/s (1210kB/s), 1181KiB/s-1181KiB/s (1210kB/s-

```

```
1210kB/s), io=34.6MiB (36.3MB), run=30008-30008msec
WRITE: bw=1220KiB/s (1249kB/s), 1220KiB/s-1220KiB/s (1249kB/s-
1249kB/s), io=35.7MiB (37.5MB), run=30008-30008msec

Disk stats (read/write):
vda: ios=8858/1027, merge=0/10, ticks=58623/68032, in_queue=126660,
util=99.75%
```

Resultados de las Pruebas de Rendimiento:

Lectura (Read): 1181 KiB/s, IOPS aproximadamente 295. Escritura (Write): 1220 KiB/s, IOPS aproximadamente 304.

Los tiempos de respuesta (latencia) y los percentiles indican un rendimiento razonable, pero siempre se puede comparar con las expectativas o requisitos específicos del sistema.

8. Analizar los datos obtenidos en los puntos 6 y 7 y extraer conclusiones sobre el rendimiento del sistema de archivo distribuido y sus prestaciones. (https://www.server-world.info/en/note?os=Debian_11&p=glusterfs9&f=4).

Punto 6 - GlusterFS sin NFS-Ganesha

En el punto 6, se utilizaron múltiples nodos para llevar a cabo pruebas de lectura y escritura concurrentes directamente en el volumen de GlusterFS sin la intermediación de NFS-Ganesha. Las pruebas demostraron un buen rendimiento en escritura (17.2 MiB/s) y lectura (5.5 MiB/s), aunque la latencia en lecturas era alta, lo que podría indicar un cuello de botella en el acceso a disco o en la red entre los nodos.

Punto 7 - GlusterFS con NFS-Ganesha

Para el punto 7, se configuró NFS-Ganesha para exportar un volumen de GlusterFS y se realizaron pruebas similares. El rendimiento observado fue significativamente menor en comparación con las pruebas directas sobre GlusterFS en el punto 6. Las velocidades de lectura y escritura fueron de aproximadamente 1.2 MiB/s, lo que es considerablemente menor que las pruebas directas.

Comparación y Análisis

- **Impacto de NFS-Ganesha:** La integración de NFS-Ganesha introduce una capa adicional de abstracción y manejo que parece tener un impacto negativo considerable en el rendimiento general del sistema de archivos. Esto es evidente en la reducción de la tasa de transferencia y el aumento de la latencia.
- **Configuración y Optimización:** Es crucial revisar la configuración de NFS-Ganesha y GlusterFS para asegurarse de que están optimizados para el tipo de cargas de trabajo y el entorno de red específicos. Las pruebas sugieren que podría haber configuraciones subóptimas que están afectando el rendimiento.
- **Capacidad de Escalabilidad:** Mientras que GlusterFS demostró una mejor capacidad de manejo bajo cargas directas, el uso de NFS-Ganesha podría estar limitando esta capacidad. Sería prudente explorar opciones de afinamiento de ambos sistemas para mejorar el rendimiento en configuraciones NFS.
- **Fiabilidad vs. Rendimiento:** Aunque NFS-Ganesha puede ofrecer ventajas en términos de flexibilidad y características de seguridad (como la autenticación y la configuración más granular de

permisos), esto parece venir a costa de un rendimiento puro.