

Cluster Disk Menggunakan Glusterfs

GlusterFS adalah sebuah sistem penyimpanan file yang dapat digunakan di beberapa server. Ia memungkinkan server untuk berbagi dan mengelola data bersama secara terdistribusi. Berikut arsitektur Glusterfs yaitu

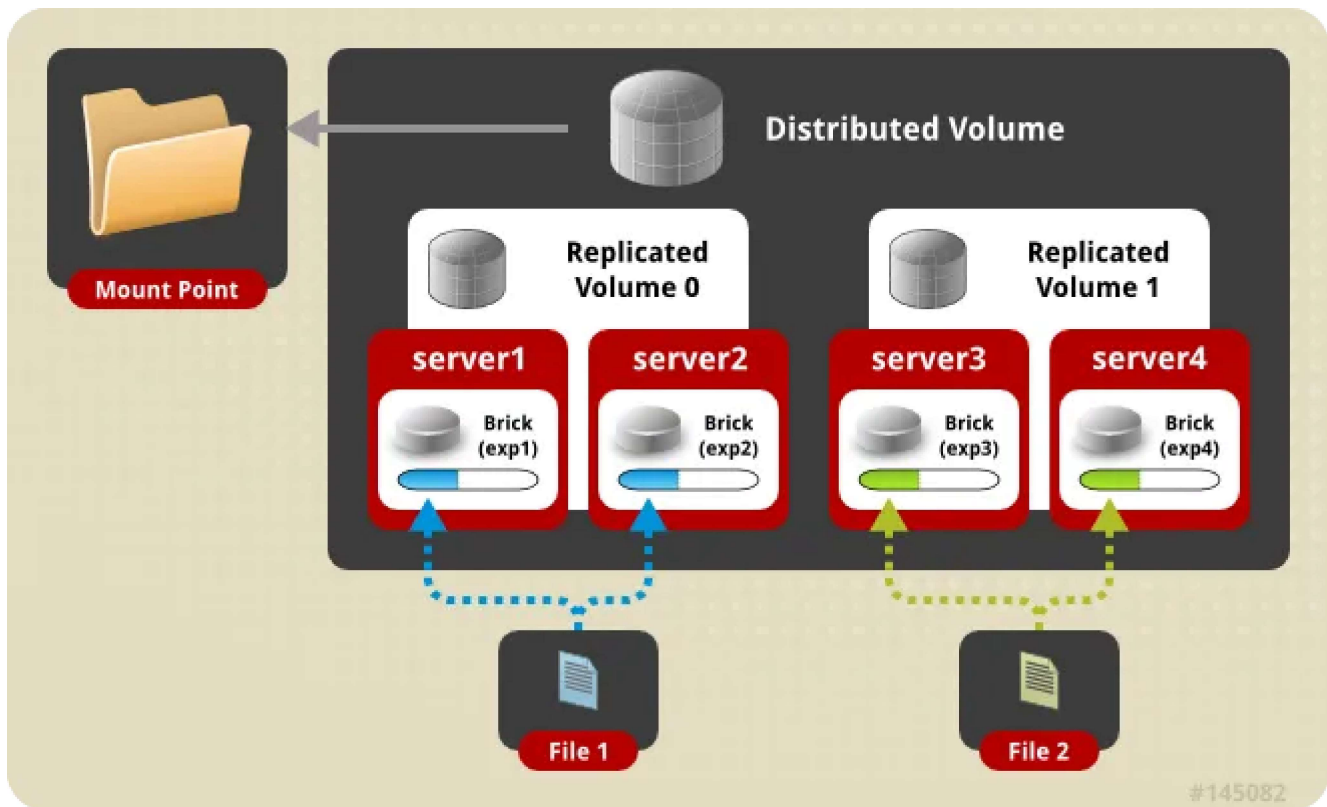
1. Distributed Glusterfs Volume

“Distributed Glusterfs Volume” — Ini adalah volume GlusterFS standar, yaitu saat Anda membuat sebuah volume tanpa menentukan jenis volumenya, pilihan defaultnya adalah membuat volume terdistribusi. Di sini, berkas-berkas tersebar di berbagai “brick” dalam volume tersebut. Jadi file1 mungkin hanya disimpan di brick1 atau brick2, tetapi tidak di keduanya. Oleh karena itu, tidak ada salinan data. Tujuan dari jenis penyimpanan volume ini adalah untuk memperbesar ukuran volume dengan mudah dan murah. Namun, ini juga berarti bahwa kegagalan pada brick akan menyebabkan hilangnya semua data, dan kita harus mengandalkan perangkat keras yang mendasarinya untuk perlindungan terhadap hilangnya data.


```
Bricks:  
Brick1: server1:/exp1  
Brick2: server2:/exp2  
Brick3: server3:/exp3  
Brick4: server4:/exp4
```

2. Replicated Glusterfs Volume

“**R**eplicated Glusterfs Volume” — Pada jenis volume ini, kita mengatasi masalah kehilangan data yang dihadapi pada volume terdistribusi. Di sini, salinan persis data dijaga di semua “brick” (unit penyimpanan). Jumlah salinan dalam volume dapat ditentukan oleh klien saat membuat volume. Jadi, kita perlu setidaknya dua “brick” untuk membuat volume dengan 2 salinan atau setidaknya tiga “brick” untuk membuat volume dengan 3 salinan. Salah satu keuntungan utama dari jenis volume seperti ini adalah bahwa bahkan jika satu “brick” mengalami kegagalan, data masih dapat diakses dari “brick” yang ada sebagai salinannya. Jenis volume seperti ini digunakan untuk meningkatkan kehandalan dan keberulangan data.



Sumber: <https://cloud.githubusercontent.com/assets/10970993/7412402/23a17eae-ef60-11e4-8813-a40a2384c5c2.png>

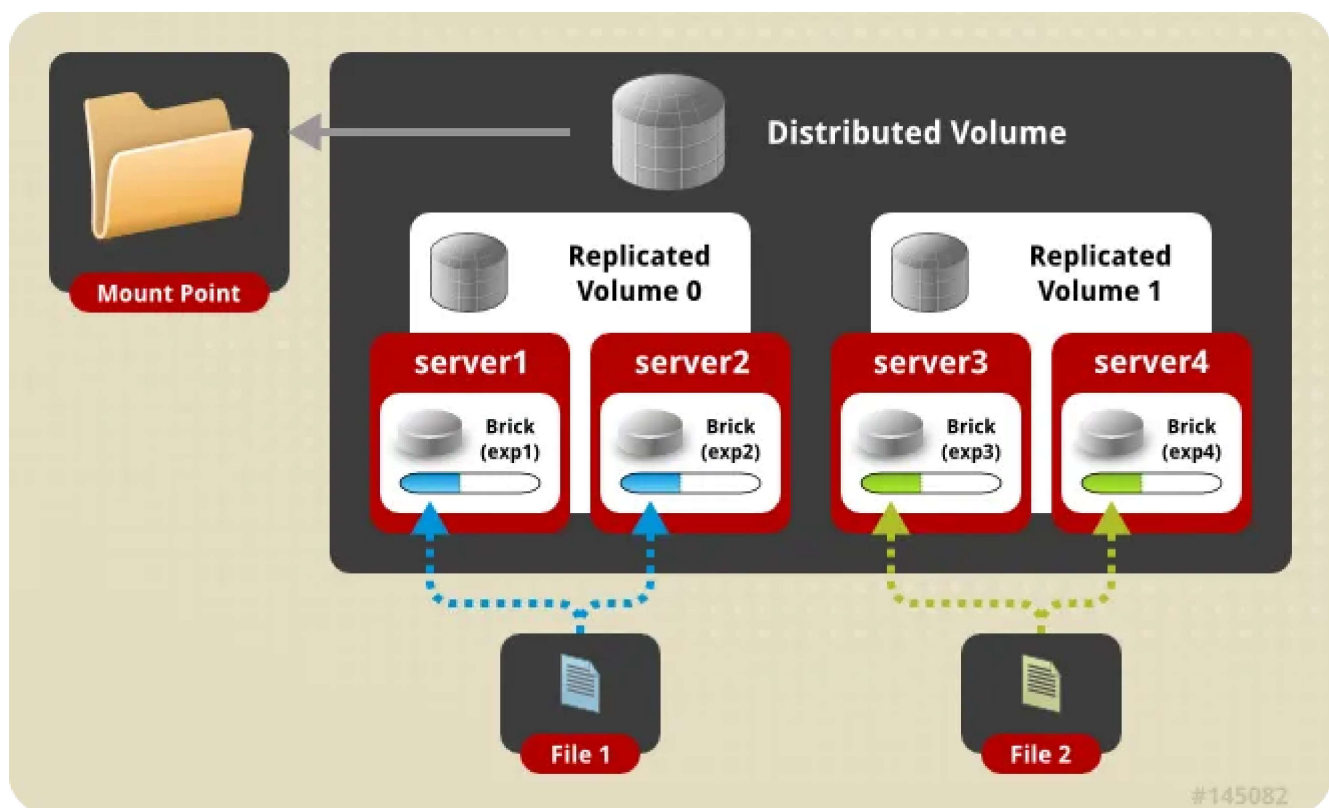
For example, four node distributed (replicated) volume with a two-way mirror:

```
# gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp2
Creation of test-volume has been successful
Please start the volume to access data
```

3. Distributed Replicated Glusterfs Volume

“Distributed Replicated Glusterfs Volume” — Pada jenis volume ini, berkas-berkas didistribusikan di antara kumpulan brick yang terduplikasi. Jumlah brick harus merupakan kelipatan dari jumlah salinan. Juga urutan di mana kita menentukan brick penting karena brick yang

bersebelahan menjadi salinan dari satu sama lain. Jenis volume ini digunakan saat diperlukan ketersediaan data yang tinggi karena redundansi dan peningkatan kapasitas penyimpanan. Jadi, jika terdapat delapan brick dan jumlah salinan adalah 2, maka dua brick pertama menjadi salinan satu sama lain, kemudian dua berikutnya, dan seterusnya. Volume ini dinotasikan sebagai 4x2. Begitu juga jika terdapat delapan brick dan jumlah salinan adalah 4, maka empat brick menjadi salinan satu sama lain, dan kita menunjuk volume ini sebagai volume 2x4.



Sumber: <https://cloud.githubusercontent.com/assets/10970993/7412402/23a17eae-ef60-11e4-8813-a40a2384c5c2.png>

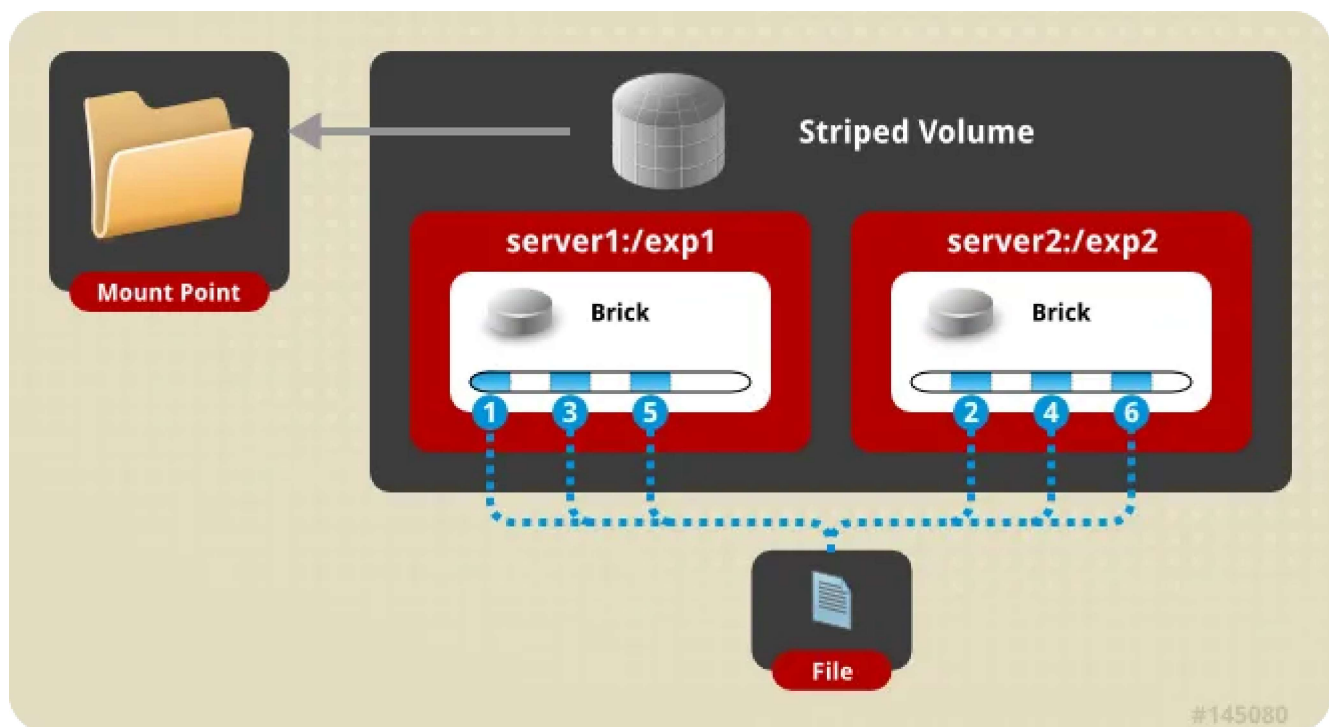
For example, four node distributed (replicated) volume with a two-way mirror:

```
# gluster volume create test-volume replica 2 transport tcp server1:/exp1 server
Creation of test-volume has been successful
```

Please start the volume to access data

4. Striped Glusterfs Volume

“**Striped Glusterfs Volume**” — Pertimbangkan berkas besar yang disimpan di dalam sebuah “brick” yang sering diakses oleh banyak klien secara bersamaan. Hal ini akan menyebabkan beban yang berlebihan pada satu “brick” dan dapat mengurangi kinerjanya. Pada **Striped Glusterfs Volume**, data disimpan di dalam “brick” setelah dibagi menjadi potongan-potongan yang berbeda. Dengan demikian, berkas besar akan dibagi menjadi potongan-potongan yang lebih kecil (jumlahnya sama dengan jumlah “brick” dalam volume) dan setiap potongan disimpan di dalam sebuah “brick.” Sekarang, beban didistribusikan dan berkas dapat diambil dengan lebih cepat, tetapi tidak ada penyediaan redundansi data.



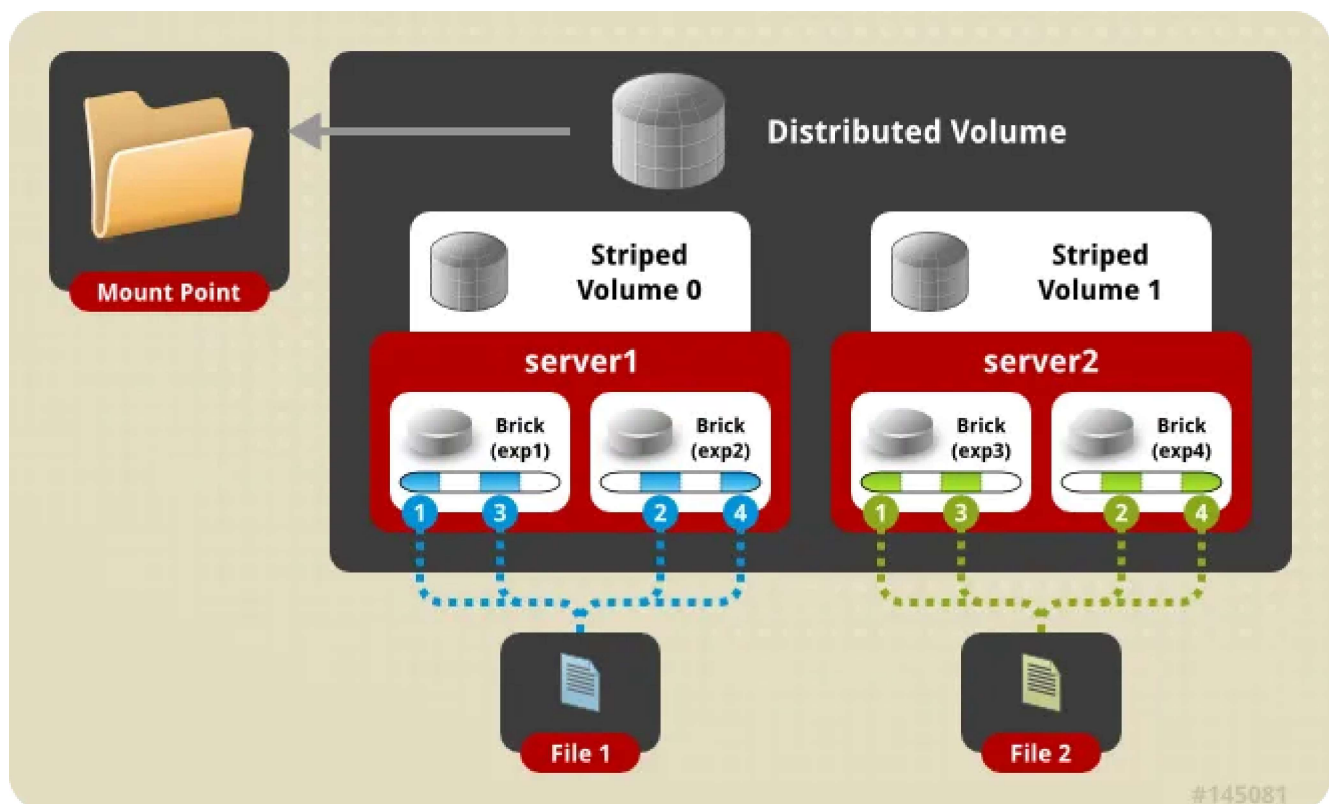
sumber: <https://cloud.githubusercontent.com/assets/10970993/7412387/f411fa56-ef5f-11e4-8e78-a0896a47625a.png>

Sebagai contoh, untuk membuat stripe volume di dua server penyimpanan:

```
# gluster volume create test-volume stripe 2 transport tcp server1:/exp1 server2  
Creation of test-volume has been successful  
Please start the volume to access data
```

5. Distributed Striped Glusterfs Volume

“Distributed Striped Glusterfs Volume” — Ini mirip dengan **Striped Glusterfs Volume**, kecuali bahwa stripe — stripe nya sekarang dapat didistribusikan di sejumlah unit penyimpanan (“brick”) yang lebih banyak. Namun, jumlah unit penyimpanan (“brick”) harus merupakan kelipatan dari jumlah stripe. Jadi, jika kita ingin meningkatkan ukuran volume, kita harus menambahkan unit penyimpanan (“brick”) dalam kelipatan jumlah stripe.



sumber: <https://cloud.githubusercontent.com/assets/10970993/7412394/0ce267d2-ef60-11e4-9959-43465a2a25f7.png>

Sebagai contoh, untuk membuat **Distributed Striped Glusterfs Volume** di delapan server penyimpanan:

```
# gluster volume create test-volume stripe 4 transport tcp  
server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4 server5:/exp5 server6:/  
Creation of test-volume has been successful  
Please start the volume to access data.
```

Latihan-1 : Implementasi **Distributed Glusterfs Volume**

Server1 (ubuntu server menggunakan virtual box)

hostname : servera

ipaddress : 172.16.67.73

Server2 (ubuntu server menggunakan virtual box)

hostname: serverb

ipaddress: 172.16.67.74

Tambahkan hardisk baru dimasing masing virtual machine

Pilih setting
Pilih storage
Pilih Controller: SCSI, klik adds hard disk
Hard Disk Selector, klik **create**

Virtual Hard disk file type pilih VDI, next sampai selesai

```
root@servera:/home/cloud4ti# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0      0 40.9M  1 loop /snap/snapd/20290
loop1        7:1      0 91.9M  1 loop /snap/lxd/24061
loop2        7:2      0 63.5M  1 loop /snap/core20/2015
sda          8:0      0  10G   0 disk
├─sda1       8:1      0  9.9G   0 part /
├─sda14      8:14     0    4M   0 part
└─sda15      8:15     0 106M   0 part /boot/efi
sdb          8:16     0  25G   0 disk
sr0         11:0     1 1024M   0 rom
```

```
root@serverb:/home/cloud4ti# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0      0 91.9M  1 loop /snap/lxd/24061
loop1        7:1      0 63.5M  1 loop /snap/core20/2015
loop2        7:2      0 40.9M  1 loop /snap/snapd/20290
sda          8:0      0  10G   0 disk
├─sda1       8:1      0  9.9G   0 part /
├─sda14      8:14     0    4M   0 part
└─sda15      8:15     0 106M   0 part /boot/efi
sdb          8:16     0  25G   0 disk
sr0         11:0     1 1024M   0 rom
```

Format Disk dikedua server

```
root@servera:/home/cloud4ti# fdisk /dev/sdb
```

Welcome to **fdisk** (**util-linux 2.34**).

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 0xac09bb52.

```

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):
First sector (2048-52428799, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-52428799, default 52428799):

Created a new partition 1 of type 'Linux' and of size 25 GiB.

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

```

```

# lsblk
sdb      8:16    0   25G    0 disk
└─sdb1    8:17    0   25G    0 part
sr0      11:0     1 1024M    0 rom

```

```

root@servera:/home/cloud4ti# mkfs.ext4 /dev/sdb1
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 6553344 4k blocks and 1638400 inodes
Filesystem UUID: a935bd77-b60b-4cfe-b93b-2baf889c346e
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

```

```

root@servera:/home/cloud4ti# mkdir /data
root@servera:/home/cloud4ti# mount /dev/sdb1 /data

```

```
# nano /etc/fstab
```

```
LABEL=cloudimg-rootfs    /          ext4    defaults    0 1
LABEL=UEFI                /boot/efi  vfat    umask=0077  0 1
/dev/sdb1                 /data      ext4    defaults    0 0
```

```
# mount -a
# systemctl daemon-reload
```

Edit /etc/hosts dikedua server

```
172.16.67.73 servera
172.16.67.74 serverb
```

Tes ping

```
root@servera:/data# ping serverb
PING serverb (172.16.67.74) 56(84) bytes of data.
64 bytes from serverb (172.16.67.74): icmp_seq=1 ttl=64 time=0.286 ms
64 bytes from serverb (172.16.67.74): icmp_seq=2 ttl=64 time=0.239 ms
64 bytes from serverb (172.16.67.74): icmp_seq=3 ttl=64 time=0.177 ms

root@serverb:/home/cloud4ti# ping servera
PING servera (172.16.67.73) 56(84) bytes of data.
64 bytes from servera (172.16.67.73): icmp_seq=1 ttl=64 time=0.184 ms
64 bytes from servera (172.16.67.73): icmp_seq=2 ttl=64 time=0.185 ms
64 bytes from servera (172.16.67.73): icmp_seq=3 ttl=64 time=0.185 ms
```

Install Glusterfs dikedua server

```
# apt install glusterfs-server -y
# systemctl enable --now glusterd
# systemctl status glusterd
```

Create gluster peer (hanya disalah satu server), pada praktikum ini kita lakukan di servera

```
root@servera:~# gluster peer probe serverb
root@servera:~# gluster peer status
Number of Peers: 1

Hostname: serverb
Uuid: 012e3431-c6fd-48c9-9af4-6fc57bc7a58f
State: Peer in Cluster (Connected)

root@serverb:/home/cloud4ti# gluster peer status
Number of Peers: 1

Hostname: servera
Uuid: fb0a436d-0363-4d02-b14c-f25a5dc0be04
State: Peer in Cluster (Connected)
```

Buat folder di /data dan create volume

```
# mkdir -p /data/gluster
# gluster volume create voldata1 servera:/data/gluster serverb:/data/gluster

root@serverb:/data# gluster volume info

Volume Name: voldata1
Type: Distribute
Volume ID: a0ca43f3-94dc-4174-919c-9f7c236bca23
Status: Created
Snapshot Count: 0
Number of Bricks: 2
```

```

Transport-type: tcp
Bricks:
Brick1: servera:/data/gluster
Brick2: serverb:/data/gluster
Options Reconfigured:
transport.address-family: inet
storage.fips-mode-rchecksum: on
nfs.disable: on

```

```

root@servera:/data# gluster volume start voldata1
volume start: voldata1: success
root@servera:/data# mount -t glusterfs servera:/voldata1 /mnt

```

```

df -Th
root@servera:/data# df -Th

```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
udev	devtmpfs	969M	0	969M	0%	/dev
tmpfs	tmpfs	198M	968K	197M	1%	/run
/dev/sda1	ext4	9.6G	1.7G	7.9G	18%	/
tmpfs	tmpfs	986M	0	986M	0%	/dev/shmcloud
tmpfs	tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	tmpfs	986M	0	986M	0%	/sys/fs/cgroup
/dev/sdb1	ext4	25G	84K	24G	1%	/data
/dev/sda15	vfat	105M	6.1M	99M	6%	/boot/efi
/dev/loop0	squashfs	41M	41M	0	100%	/snap/snapd/20290
/dev/loop1	squashfs	64M	64M	0	100%	/snap/core20/2015
/dev/loop2	squashfs	92M	92M	0	100%	/snap/lxd/24061
tmpfs	tmpfs	198M	0	198M	0%	/run/user/1004
servera:/voldata1	fuse.glusterfs	49G	501M	47G	2%	/mnt

Create file

```

# cd /mnt
# for i in {1..28}
do

```

```
touch file$i.txt  
done
```



Written by Harumin

4 Followers

Menulislah agar engkau dikenang orang

Follow



More from Harumin



 Harumin

Chatting Server using Openfire

Openfire adalah sebuah server pesan instan dan kolaborasi berbasis protokol XMPP...



 Harumin

Openstack using mickrostack

Openstack adalah sebuah platform cloud computing yang dirancang untuk...