

TUTORIAL FOR DUMMIES

GLUSTER FILE SYSTEM SU UBUNTU 20.04 LTS

Appunti di Pietro Cornelio

Rel. doc. 0.1.0 del 09 Luglio 2020

Rel. doc. 0.1.1 del 19 Marzo 2022 [verificata compatibilità con Ubuntu 22.04 LTS]

Rel. doc. 0.1.2 del 25 Marzo 2023 [aggiunte altre info]

Rel. doc. 0.1.3 del 27 Marzo 2023 [osservazioni su volumi striped]

Rel. doc. 0.1.4 del 21 Aprile 2023 [correzione errori mount volumi in /etc/fstab]



Prefazione

GlusterFS è un file system open source distribuito e scalabile che permette la memorizzazione di risorse su dispositivi di archiviazione in una rete.

GlusterFS gestisce fino a diversi petabyte di dati, gestisce migliaia di client e li rende accessibili su Infiniband RDMA (remote direct memory access, fibra ottica) o connessioni TCP/IP.

Le risorse che successivamente vengono messe in condivisione possono essere montate dai client attraverso i protocolli CIFS, NFS oppure attraverso il client nativo Gluster.

Inoltre GlusterFS supporta la replica geografica ossia la replica dei dati di un Volume su un Server dislocato in un area geografica diversa da quella dove sono presenti gli altri nodi.

Qui tutti i riferimenti tecnici: [Gluster](https://www.gluster.org/)

Sommario

<i>TUTORIAL FOR DUMMIES</i>	1
<i>GLUSTER FILE SYSTEM SU UBUNTU 20.04 LTS</i>	1
Prefazione	1
Introduzione	3
Funzionamento di GlusterFS.....	3
Installazione.....	4
Tipologia di volume di cluster storage.....	5
Distributed Volume.....	5
Replicated Volume.....	6
Distributed Replicated Volume	6
Striped Volume	7
Distributed Striped Volume	7
Schema infrastrutturale per questo impianto	8
Creazione volume	9
Creazione Disco	9
Installiamo Glusterfs server	10
Configurare il Volume GlusterFS.....	10
Creiamo il volume replica su 3 nodi.....	11
Installiamo GlusterFS client sui nodi client.....	12
Creiamo le directory che dovranno contenere le applicazioni dal nodo client.....	13
settare l'orario corretto.....	13
Sicurezza dei mount clients	13
Vediamo lo spazio occupato dal volume	13
Gestire le quota delle directory sotto volume Gluster	14
Tuning del kernel Linux.....	15
Appunti e info	16
Cosa sono i bricks (mattoni)	16
Verifica delle porte di networking usate da Gluster	16
Accesso ai dati - Configurazione del client GlusterFS	17
Montaggio dei volumi Gluster	19
Protocollo NFS (Network File System)	21
Protocollo MS Windows CIFS e SAMBA.....	23
NFS-Ganesha su GlusterFS.....	25

Introduzione

Il nome “Gluster” deriva da “**GNU**” (**GNU’s not Unix**) e “**Cluster**”. Il sistema è stato pubblicato sotto la GNU General Public License (GNU-GPLS) e può quindi essere utilizzato gratuitamente.

Il progetto è stato pubblicato nel 2005 da Gluster Inc. Nel 2011, il distributore Linux RedHat ha rilevato l’azienda e da allora ha continuato a sviluppare il file system Gluster.

Funzionamento di GlusterFS

Un file system distribuito ha senso solo se più computer sono collegati tra loro. Nella documentazione di GlusterFS vengono richiesti almeno tre server. Il termine “server”, tuttavia, non va preso alla lettera, in quanto può comprendere praticamente ogni forma di hardware fisico o emulato. Oltre a qualsiasi computer, è fattibile anche l’uso di macchine virtuali, con vantaggi soprattutto in termini di flessibilità.

I server integrati fungono da Nodes (punti nodali), che si collegano tramite la rete TCP/IP. I dispositivi integrati formano un cosiddetto **Trusted Pool**, ovvero un’interconnessione di server affidabili. Questi mettono a disposizione la loro memoria sotto forma di **Bricks (mattoni)**, sui quali vengono formati i **Volumes (unità)**. Questi ultimi possono essere montati e utilizzati come normali supporti dati. I computer che accedono vengono definiti client. Un nodo può essere sia server, sia client.

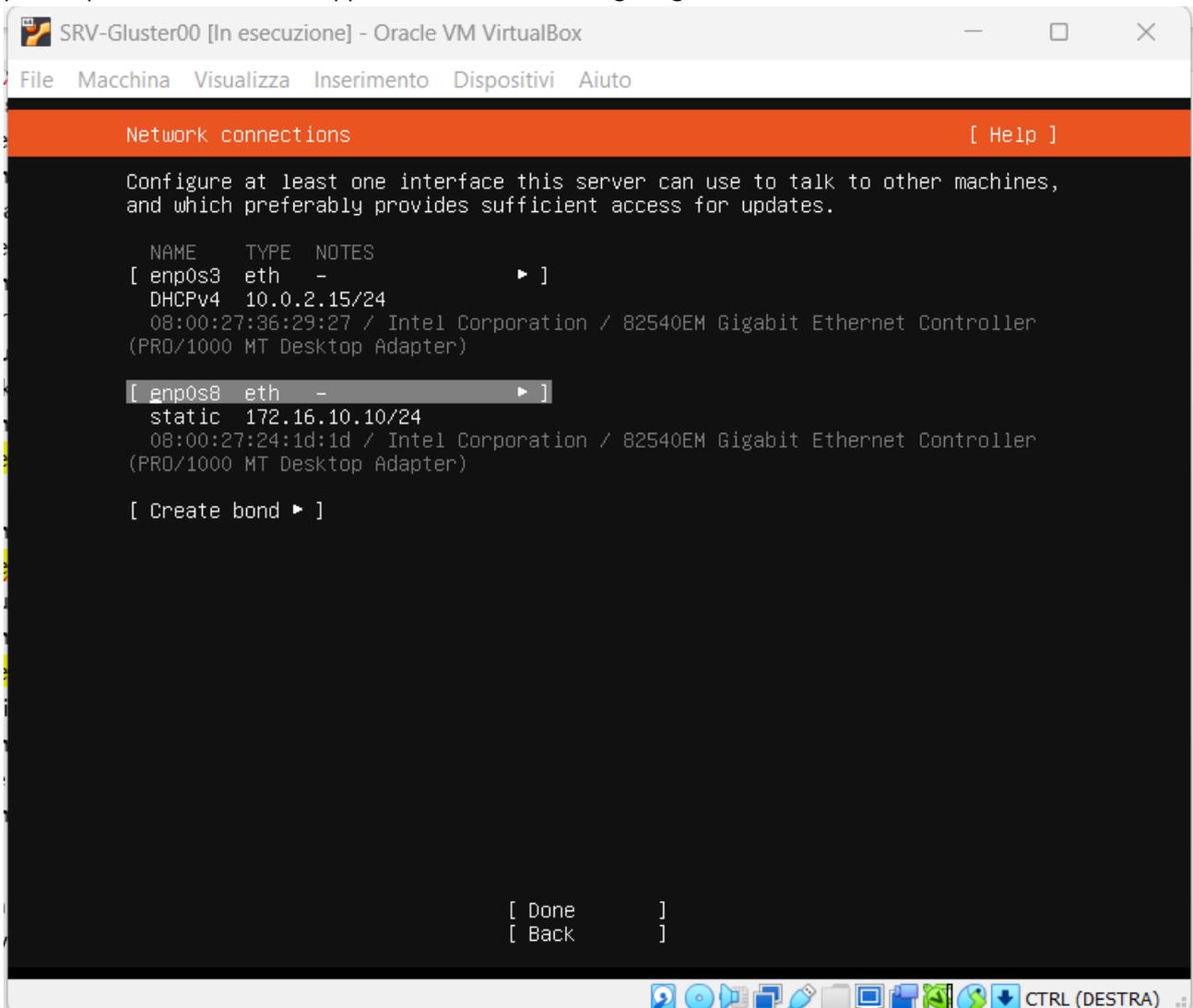
Una caratteristica particolare è la grande scalabilità: infatti è possibile aggiungere in un secondo momento un numero qualsiasi di Nodes e Bricks. Ciò consente di adattare continuamente le dimensioni dello spazio di memoria in base alle esigenze. La dimensione massima della memoria da gestire è di diversi petabyte. Inoltre, GlusterFS garantisce la sicurezza attraverso la ridondanza. Il rischio di guasto è inizialmente distribuito su più sistemi, che possono anche essere separati fisicamente. Inoltre, è possibile impostare gruppi RAID, dove, invece del Volume Distribuito standard, deve essere creato un Volume Replicato. Questo volume memorizza ogni file due volte e corrisponde ad un Mirror-RAID.

Installazione

Nel mio caso ho usato Oracle Virtual Box ed in particolare 3 nodi server come segue:

2 CPU – 2 Gb RAM - 2 HD (1 da 16 Gb dedicato all'OS montato su radice / e 1 da 20 Gb dedicato al volume Gluster montato su /GlusterVolume)

2 Schede di rete: una NIC in modalità NAT (ip statici) verso l'esterno (Internet), mentre l'altra NIC interna privata per le comunicazioni applicative e di client storage. Ogni NIC su 2 sottoreti diverse:



```
SRV-Gluster00 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
Network connections [ Help ]
Configure at least one interface this server can use to talk to other machines,
and which preferably provides sufficient access for updates.

NAME    TYPE  NOTES
[ enp0s3 eth - ]
DHCPv4  10.0.2.15/24
08:00:27:36:29:27 / Intel Corporation / 82540EM Gigabit Ethernet Controller
(PRO/1000 MT Desktop Adapter)

[ enp0s8 eth - ]
static  172.16.10.10/24
08:00:27:24:1d:1d / Intel Corporation / 82540EM Gigabit Ethernet Controller
(PRO/1000 MT Desktop Adapter)

[ Create bond ]

[ Done ]
[ Back ]
```

Poi un 4° nodo come client Gluster.

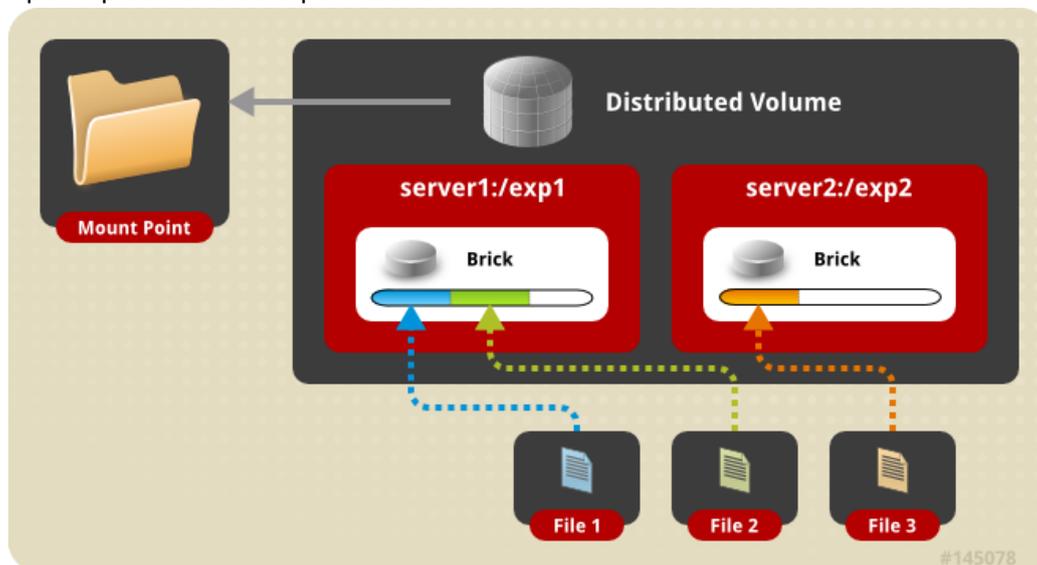
Tipologia di volume di cluster storage

Con gluster esistono 5 tipi di volumi di storage cluster specifiche per assolvere a ogni esigenza.

- 1) Distributed Volume
- 2) Replicated Volume
- 3) Distributed Replicated Volume
- 4) Striped Volume
- 5) Distributed Striped Volume

Distributed Volume

Questo è il volume glusterfs predefinito, ovvero, durante la creazione di un volume se non si specifica il tipo di volume, l'opzione predefinita è creare un volume distribuito. Qui, i file sono distribuiti su vari mattoni (chiamati bricks) nel volume. Quindi file1 può essere memorizzato solo in brick1 o brick2 ma non su entrambi. Quindi non c'è ridondanza dei dati. Lo scopo di un tale volume di archiviazione è quello di ridimensionare facilmente ed economicamente le dimensioni del volume. Tuttavia, ciò significa anche che un errore di un mattone porterà alla completa perdita di dati e si deve fare affidamento sull'hardware sottostante per la protezione dalla perdita di dati.



Per questa tipologia di volume, bisogna osservare la best practice, in un volume replicato di GlusterFS non ci sono conflitti di scrittura quando il percorso (path) del brick è lo stesso su tutti i nodi, poiché tutti i nodi contengono una copia completa dei dati. Tuttavia, nel caso di un **volume a strisce (striped)** di GlusterFS, la situazione è diversa perché i dati vengono suddivisi in blocchi e distribuiti tra i nodi, in modo che la capacità di archiviazione complessiva del volume sia la somma delle capacità di ogni nodo.

In questo caso, se due o più nodi cercano di scrivere contemporaneamente nello stesso blocco di dati, ci possono essere conflitti che potrebbero causare la corruzione dei dati. Quindi, in un volume striped di GlusterFS, **è importante evitare di creare una directory con lo stesso percorso su tutti i nodi.**

Pertanto con un volume striped, è importante evitare di creare una directory con lo stesso percorso su tutti i nodi. Ad esempio se abbiamo 3 nodi che compongono lo "Storage Pool" si creano le seguenti directory:

/data/brick1 (su nodo1)

/data/brick2 (su nodo2)

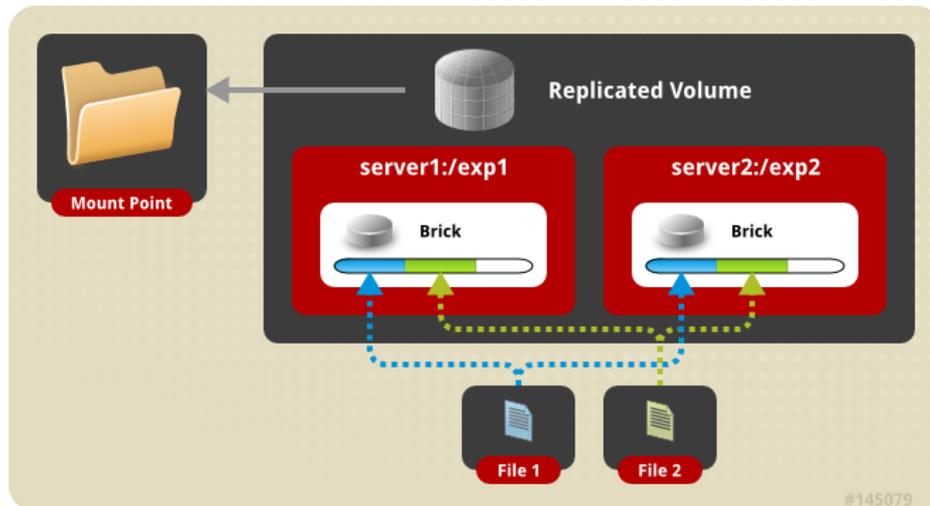
/data/brick3 (su nodo3)

Di seguito l'esempio per creare il volume

```
sudo gluster volume create GlusterVolStriped01 replica 3 transport tcp \  
srv-gluster00:/data/brick1 \  
srv-gluster01:/data/brick2 \  
srv-gluster02:/data/brick3 force
```

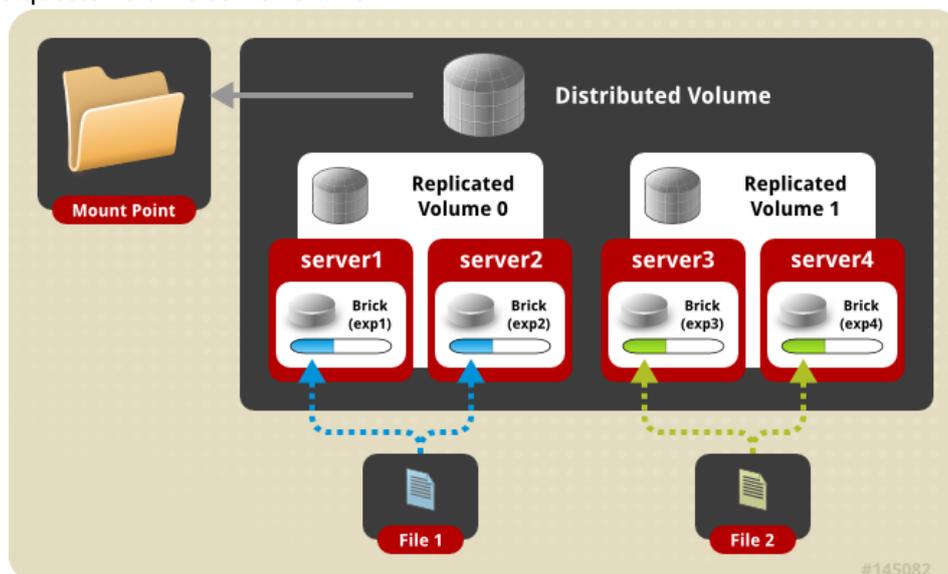
Replicated Volume

In questo volume superiamo il problema della perdita di dati affrontato nel volume distribuito. Qui vengono mantenute copie esatte dei dati su tutti i mattoni. Il numero di repliche nel volume può essere deciso dal client durante la creazione del volume. Quindi abbiamo bisogno di almeno due mattoni per creare un volume con 2 repliche o un minimo di tre mattoni (bricks) per creare un volume di 3 repliche. Uno dei principali vantaggi di un tale volume è che anche se un mattone si guasta, è comunque possibile accedere ai dati dai suoi mattoni replicati. **Tale volume viene utilizzato per una migliore affidabilità e ridondanza dei dati.**



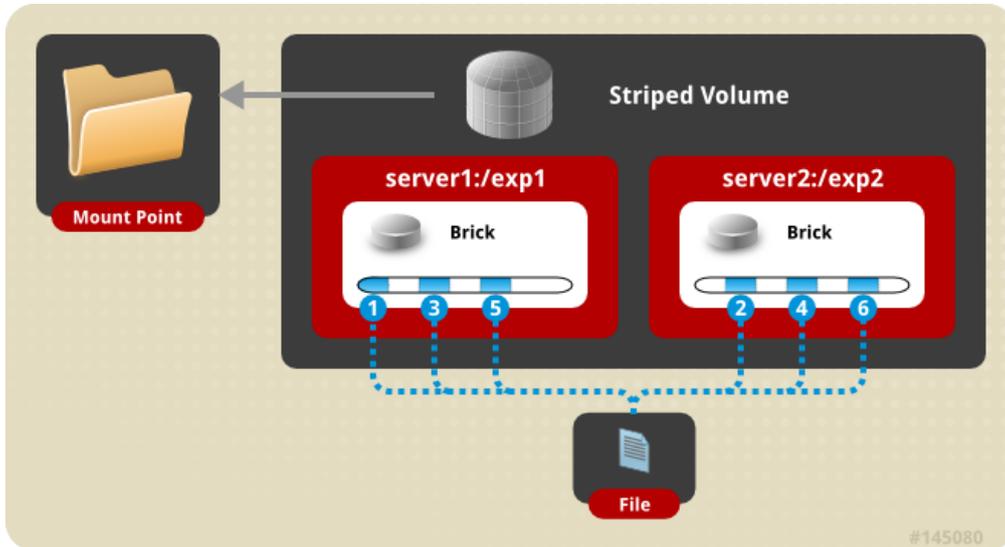
Distributed Replicated Volume

In questo volume i file sono distribuiti su set di mattoni replicati. Il numero di brick deve essere un multiplo del numero di repliche. Anche l'ordine in cui specifichiamo i mattoni è importante poiché i mattoni adiacenti diventano repliche l'uno dell'altro. Questo tipo di volume viene utilizzato quando è richiesta un'elevata disponibilità dei dati a causa della ridondanza e dello storage di scalabilità. Quindi, se ci sono otto mattoncini e le repliche contano 2, i primi due mattoncini diventano repliche l'uno dell'altro, poi i due successivi e così via. Questo volume è indicato come 4x2. Allo stesso modo, se ci fossero otto mattoni e il numero di repliche fosse 4, allora quattro mattoni diventerebbero una replica l'uno dell'altro e indicheremmo questo volume come volume 2x4.



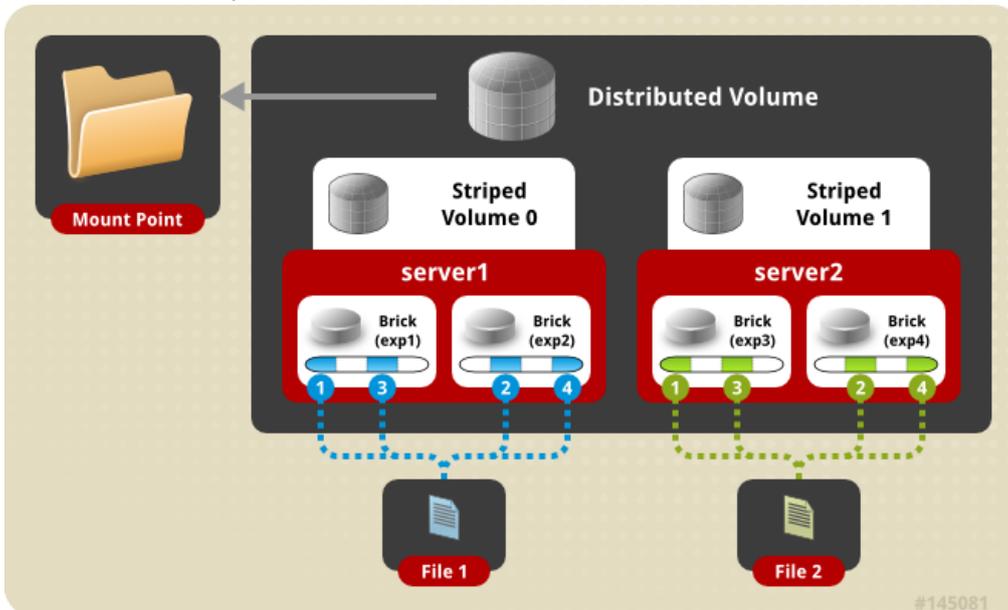
Striped Volume

Si consideri un file di grandi dimensioni archiviato in un mattone a cui accedono frequentemente molti client contemporaneamente. Ciò causerà un carico eccessivo su un singolo mattone e ridurrebbe le prestazioni. Nel volume striped i dati vengono archiviati nei mattoni dopo averli divisi in diverse stripes (strisce). Quindi il file di grandi dimensioni verrà diviso in blocchi più piccoli (pari al numero di mattoni nel volume) e ogni blocco viene memorizzato in un mattone. Ora il carico è distribuito e il file può essere recuperato più velocemente ma non viene fornita alcuna ridondanza dei dati.



Distributed Striped Volume

Questo è simile al volume Striped Glusterfs tranne per il fatto che le stripes possono ora essere distribuite su un numero maggiore di mattoni. Tuttavia, il numero di mattoni deve essere un multiplo del numero di stripes (strisce). Quindi, se vogliamo aumentare la dimensione del volume, dobbiamo aggiungere mattoni nel multiplo del numero di stripes.



Schema infrastrutturale per questo impianto

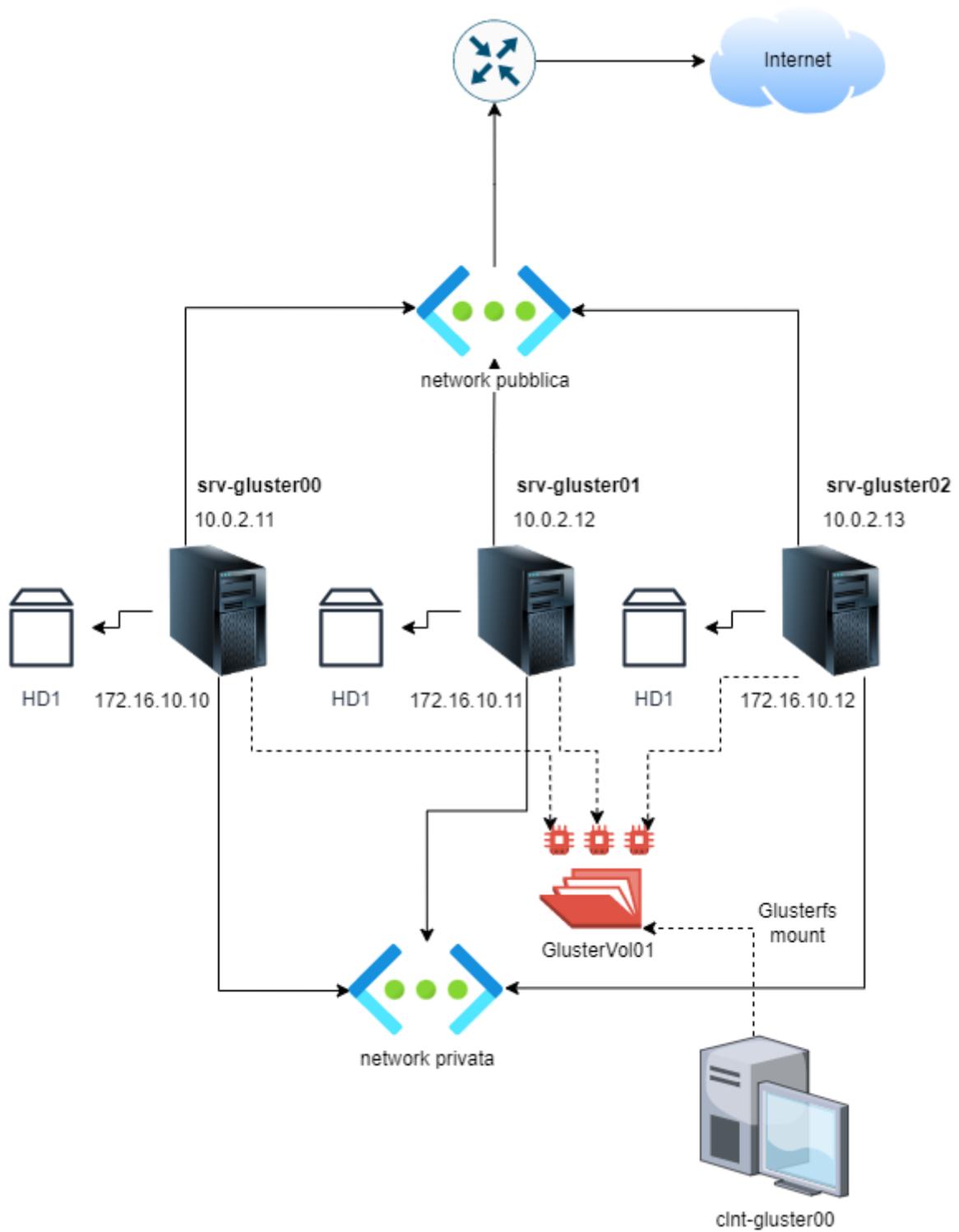


Figura 1 - schema infrastrutturale

Creazione volume

Dedicare un 2° disco formattato XFS (o altro file system ad alte prestazioni come ZFS o Btrfs) e montarlo con un nome esplicativo, ad esempio come nel nostro tutorial evidenziato sotto in giallo:

```
pietro@srv-gluster00:~$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
tmpfs	198M	1.1M	197M	1%	/run
/dev/sda2	16G	5.0G	10G	34%	/
tmpfs	988M	0	988M	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sdb1	20G	175M	20G	1%	/GlusterVolume
tmpfs	198M	4.0K	198M	1%	/run/user/1000

In questo esempio, il file system del 1° disco dedicato al sistema operativo, non è stato partizionato secondo le best practices (partition scheme), ad esempio come i seguenti mount separati:

```
/boot  
/var  
/home  
/tmp  
/usr
```

Perché si tratta di un nodo da dedicare a questo tutorial che non dovrà assolvere a compiti massivi di servizio storage. Per il nostro tutorial, considerando ci interessa l'HA creeremo un volume replicated.

Creazione Disco

```
sudo fdisk /dev/sdb
```

creare una nuova partizione:

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

Formattiamo la nuova partizione

```
sudo mkfs.xfs /dev/sdb1  
creiamo la directory dove montare la nuova partizione  
sudo mkdir /GlusterVolume
```

montiamo la nuova partizione

```
sudo mount -t xfs /dev/sdb1 /GlusterVolume
```

verifichiamo la partizione montata è scrivibile

```
sudo touch /GlusterVolume/pippo.txt  
se la partizione è scrivibile, smontiamola con: sudo umount /GlusterVolume
```

modifichiamo /etc/fstab

```
sudo vim /etc/fstab  
aggiungendo: /dev/sdb1 /glustervolume xfs defaults 0 0
```

facciamo una prova del mount automatico

```
mount -a
```

Installiamo Glusterfs server

```
sudo apt-get install glusterfs-server -y
```

Avviamo e abilitiamo i servizi glusterfs

```
sudo systemctl start glusterd
sudo systemctl enable glusterd
```

verifichiamo che il servizio sia Up & Running

```
sudo systemctl status glusterd
```

Ripetiamo l'installazione dei pacchetti per gli altri 2 nodi

Nel nostro caso realizzeremo uno storage ad alta affidabilità in HA composto da **3 nodi server Gluster**.

Aggiungere nel file **/etc/hosts** le seguenti voci:

```
# Nodi GlusterFS in Replica
172.16.10.10    srv-gluster00
172.16.10.11    srv-gluster01
172.16.10.12    srv-gluster02
```

Verificare se il firewall è attivo

```
pietro@srv-gluster00:~$ sudo ufw status
Status: inactive
Altrimenti disabilitarlo con: sudo ufw disable
```

Configurare il Volume GlusterFS

Innanzitutto, sarà necessario **creare uno storage pool attendibile** aggiungendo i nodi al pool.

Per fare ciò, esegui i seguenti comandi sul nodo **srv-gluster00** :

```
sudo gluster peer probe srv-gluster00
sudo gluster peer probe srv-gluster01
sudo gluster peer probe srv-gluster02
pietro@srv-gluster00:~$ sudo gluster peer probe srv-gluster00
peer probe: Probe on localhost not needed
```

Nota: sul 1° nodo non è necessario aggiungere se stesso (evidenziato in verde)

```
pietro@srv-gluster00:~$ sudo gluster peer probe srv-gluster01
peer probe: success
pietro@srv-gluster00:~$ sudo gluster peer probe srv-gluster02
peer probe: success
```

```
pietro@srv-gluster00:~$ sudo gluster peer status
Number of Peers: 2
```

```
Hostname: srv-gluster01
Uuid: 57e544b9-df19-4082-b271-8b6957d4b12f
State: Peer in Cluster (Connected)
```

```
Hostname: srv-gluster02
Uuid: 63a652ac-e92a-4d0a-83c2-f9a0ddb6aca9
State: Peer in Cluster (Connected)
```

```
pietro@srv-gluster00:~$ sudo gluster pool list
UUID                               Hostname      State
57e544b9-df19-4082-b271-8b6957d4b12f  srv-gluster01  Connected
63a652ac-e92a-4d0a-83c2-f9a0ddb6aca9  srv-gluster02  Connected
a2fdb5ce-d015-44da-a6ed-2848e3a6bdc3  localhost      Connected
```

Diamo lo stesso comando sugli altri 2 nodi per vedere i risultati e comprendere lo Storage Pool appena creato e validato:

```

pietro@srv-gluster01:~$ sudo gluster pool list
[sudo] password for pietro:
UUID                               Hostname      State
a2fdb5ce-d015-44da-a6ed-2848e3a6bdc3  srv-gluster00  Connected
63a652ac-e92a-4d0a-83c2-f9a0ddb6aca9  srv-gluster02  Connected
57e544b9-df19-4082-b271-8b6957d4b12f  localhost      Connected

```

```

pietro@srv-gluster02:~$ sudo gluster pool list
[sudo] password for pietro:
UUID                               Hostname      State
a2fdb5ce-d015-44da-a6ed-2848e3a6bdc3  srv-gluster00  Connected
57e544b9-df19-4082-b271-8b6957d4b12f  srv-gluster01  Connected
63a652ac-e92a-4d0a-83c2-f9a0ddb6aca9  localhost      Connected

```

Creiamo il volume replica su 3 nodi

Sul 1° nodo dare il seguente impianto di comandi:

```

sudo gluster volume create GlusterVol01 replica 3 transport tcp srv-
gluster00:/GlusterVolume srv-gluster01:/GlusterVolume srv-gluster02:/GlusterVolume
force

```

```

volume create: GlusterVol01: success: please start the volume to access data

```

Verifichiamo il volume Gluster appena creato

```

pietro@srv-gluster00:~$ sudo gluster volume list
GlusterVol01

```

Ora possiamo avviare il volume Gluster denominato GlusterVol01

```

pietro@srv-gluster00:~$ sudo gluster volume start GlusterVol01
volume start: GlusterVol01: success

```

Ora visualizziamo le info del volume Gluster avviato

```

pietro@srv-gluster00:~$ sudo gluster volume info GlusterVol01

```

```

Volume Name: GlusterVol01
Type: Replicate
Volume ID: 2d206056-90a0-4323-b2dc-536a4c32457e
Status: Started
Snapshot Count: 0
Number of Bricks: 1 x 3 = 3
Transport-type: tcp
Bricks:
Brick1: srv-gluster00:/GlusterVolume
Brick2: srv-gluster01:/GlusterVolume
Brick3: srv-gluster02:/GlusterVolume
Options Reconfigured:
cluster.granular-entry-heal: on
storage.fips-mode-rchecksum: on
transport.address-family: inet
nfs.disable: on
performance.client-io-threads: off

```

Ora verifichiamo lo stato del Volume Gluster

```

pietro@srv-gluster00:~$ sudo gluster volume status GlusterVol01
Status of volume: GlusterVol01

```

Gluster process	TCP Port	RDMA Port	Online	Pid
Brick srv-gluster00:/GlusterVolume	56804	0	Y	1801
Brick srv-gluster01:/GlusterVolume	53612	0	Y	1627
Brick srv-gluster02:/GlusterVolume	50620	0	Y	1643
Self-heal Daemon on localhost	N/A	N/A	Y	1818
Self-heal Daemon on srv-gluster02	N/A	N/A	Y	1660
Self-heal Daemon on srv-gluster01	N/A	N/A	Y	1644

Task Status of Volume GlusterVol01

There are no active volume tasks

Nota:

i parametri di rete ed altre opzioni sono state lasciate quelle di default

```
pietro@srv-gluster00:/etc/glusterfs$ cat glusterd.vol
volume management
  type mgmt/glusterd
  option working-directory /var/lib/glusterd
  option transport-type socket
  option transport.socket.keepalive-time 10
  option transport.socket.keepalive-interval 2
  option transport.socket.read-fail-log off
  option transport.socket.listen-port 24007
  option ping-timeout 0
  option event-threads 1
# option lock-timer 180
# option transport.address-family inet6
# option base-port 49152
  option max-port 60999
end-volume
```

Installiamo GlusterFS client sui nodi client

Aggiungiamo le entry in /etc/hosts con

```
sudo vim /etc/hosts
# Nodi GlusterFS in Replica
172.16.10.10    srv-gluster00
172.16.10.11    srv-gluster01
172.16.10.12    srv-gluster02
```

Ora installiamo il pacchetto Client di Glusterfs

```
sudo apt-get install glusterfs-client
```

Creiamo la directory di mount

```
sudo mkdir -p /mnt/glusterfs
```

Montiamo il volume Gluster

```
sudo mount -t glusterfs srv-gluster00:/GlusterVol01 /mnt/glusterfs/
df -h
```

modifichiamo fstab per il mount automatico

```
sudo vim /etc/fstab
```

Nota: l'opzione `_netdev` viene inserita solo per i nodi clients

```
# Mount del volume GlusterVol01 del clusterfs storage replicsto su 3 nodi
srv-gluster00:/GlusterVol01 /mnt/glusterfs glusterfs defaults,_netdev 0 0 i
```

Creiamo le directory che dovranno contenere le applicazioni dal nodo client

Le seguenti directory sono di esempio e stanno a semplificare 3 dir che ospiteranno diverse esigenze funzionali che devono rimanere in HA. Da ogni nodo client specifico (nel nostro esempio 3 client) che dovrà utilizzare quella specifica cartella creare le dir:

```
pietro@clnt-gluster00:/$ sudo mkdir -p /mnt/glusterfs/database
pietro@clnt-gluster01:/$ sudo mkdir -p /mnt/glusterfs/app01
pietro@clnt-gluster02:/$ sudo mkdir -p /mnt/glusterfs/fileserver
```

assegnare i corretti privilegi di utenza alle 3 directory

serve affinché gli utenti owner delle applicazioni e dei files potranno leggere e scrivere

```
pietro@clnt-gluster00:/$ sudo chown pietro.pietro /mnt/glusterfs/database
pietro@clnt-gluster01:/$ sudo chown pippo.pippo /mnt/glusterfs/app01/
pietro@clnt-gluster02:/$ sudo chown pluto.pluto /mnt/glusterfs/fileserver/
```

settare l'orario corretto

```
sudo timedatectl set-timezone Europe/Rome
```

aggiungere i times server italiani

```
sudo vim /etc/systemd/timesyncd.conf
[Time]
NTP=ntp1.inrim.it
sudo systemctl start systemd-timesyncd.service
sudo systemctl enable systemd-timesyncd.service
```

verifichiamo

```
pietro@srv-gluster02:~$ timedatectl
Local time: Sat 2023-03-25 20:08:38 CET
Universal time: Sat 2023-03-25 19:08:38 UTC
RTC time: Sat 2023-03-25 19:08:35
Time zone: Europe/Rome (CET, +0100)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

Sicurezza dei mount clients

Ora settiamo gli IP o le sottoreti che possono collegarsi allo Storage Pool ed in particolare al volume Gluster GlusterVol01 in modo che nessun altro estraneo nodo potrà montare il volume.

Esempio:

autorizziamo solo 3 nodi a potersi collegare al volume Gluster

```
sudo gluster volume set GlusterVol01 auth.allow 172.16.10.100, 172.16.10.101,
172.16.10.103
```

Vediamo lo spazio occupato dal volume

```
df -hTP /mnt/GlusterVolume
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
srv-gluster00:/GlusterVol01	fuse.glusterfs	20G	380M	20G	2%	/mnt/glusterfs

Gestire le quota delle directory sotto volume Gluster

Visulizziamo le quota, non ci sarà nulla configurato, naturalmente:

```
pietro@srv-gluster00:~$ sudo gluster volume quota GlusterVol01 list
quota command failed : Quota is disabled, please enable quota
```

infatti risulta disabilitato

Abilitiamo le quota del volume

```
pietro@srv-gluster00:~$ sudo gluster volume quota GlusterVol01 enable
volume quota : success
```

Fissiamo una quota di 5 Gb alla directory database

Path directory /GlusterVolume/database/

Notare nell'impianto di comandi non si mette il path assoluto

```
pietro@srv-gluster00:~$
sudo gluster volume quota GlusterVol01 limit-usage /database 5GB
volume quota : success
```

verifichiamo la quota appena settata

```
pietro@srv-gluster00:~$ sudo gluster volume quota GlusterVol01 list
```

Path	Hard-limit	Soft-limit	Used	Available	Soft-limit exceeded?	Hard-limit exceeded?
/database	5.0GB	80%(4.0GB)	0Bytes	5.0GB	No	No

Fissiamo un'altra quota per la dir fileserver

```
pietro@srv-gluster00:~$
sudo gluster volume quota GlusterVol01 limit-usage /fileserver 10GB
volume quota : success
```

```
pietro@srv-gluster00:~$ sudo gluster volume quota GlusterVol01 list
```

Path	Hard-limit	Soft-limit	Used	Available	Soft-limit exceeded?	Hard-limit exceeded?
/database	5.0GB	80%(4.0GB)	0Bytes	5.0GB	No	No
/fileserver	10.0GB	80%(8.0GB)	0Bytes	10.0GB	No	No

Per disabilitare le quota al Volume Gluster

```
sudo gluster volume quota GlusterVol01 disable
```

Tuning del kernel Linux

un esempio di come potresti ottimizzare il kernel Linux per nodi Gluster:

1. **Aggiornare il kernel:** Assicurati di avere l'ultima versione del kernel Linux installata sul tuo nodo Gluster.
2. **Disabilitare lo swapping:** Disabilita lo swapping, poiché lo swapping può causare una diminuzione delle prestazioni del cluster Gluster.
3. **Impostare il valore di sched_migration_cost:** Imposta il valore di `sched_migration_cost` al valore massimo (65535), in modo che il kernel Linux riduca il numero di migrazioni del processo tra i core della CPU.
4. **Impostare il valore di vm.swappiness:** Imposta il valore di `vm.swappiness` a 0, in modo che il kernel Linux non tenti di swapare la memoria in caso di esaurimento della memoria fisica.
5. **Aumentare il valore di net.core.somaxconn:** Aumenta il valore di `net.core.somaxconn`, in modo che il kernel Linux accetti un maggior numero di connessioni TCP.
6. **Impostare il valore di net.ipv4.tcp_max_syn_backlog:** Imposta il valore di `net.ipv4.tcp_max_syn_backlog` al massimo, in modo che il kernel Linux accetti un maggior numero di connessioni TCP.
7. **Impostare il valore di net.core.netdev_max_backlog:** Imposta il valore di `net.core.netdev_max_backlog` a un valore superiore, in modo da aumentare la capacità di bufferizzazione del kernel Linux.
8. **Impostare il valore di net.ipv4.tcp_window_scaling:** Imposta il valore di `net.ipv4.tcp_window_scaling` su 1, in modo che il kernel Linux utilizzi la finestra di scaling TCP per migliorare le prestazioni della rete.
9. **Impostare il valore di net.ipv4.tcp_sack:** Imposta il valore di `net.ipv4.tcp_sack` su 1, in modo che il kernel Linux utilizzi l'opzione SACK per migliorare le prestazioni della rete.
10. **Impostare il valore di net.ipv4.tcp_timestamps:** Imposta il valore di `net.ipv4.tcp_timestamps` su 1, in modo che il kernel Linux utilizzi l'opzione TCP timestamp per migliorare le prestazioni della rete.

E' importante effettuare un backup delle impostazioni del kernel esistenti prima di apportare modifiche significative se il nodo server è già attivo con altri servizi, cioè non è stato creato da scratch.

Appunti e info

Cosa sono i bricks (mattoni)

Un brick in Gluster è un termine che si riferisce a una unità di spazio di archiviazione locale che fa parte di un cluster di archiviazione distribuito GlusterFS. In altre parole, un brick è un nodo di archiviazione individuale all'interno del cluster Gluster che contribuisce alla capacità totale di archiviazione del cluster.

Un brick Gluster è tipicamente un'unità di archiviazione locale, ad esempio un'unità disco rigido o una partizione di un disco rigido, che è montata su un nodo del cluster Gluster e che viene poi utilizzata per fornire uno spazio di archiviazione distribuito su tutta la rete.

Ad esempio, i brick Gluster vengono replicati tra i nodi del cluster per garantire la disponibilità dei dati. Ciò significa che i dati memorizzati su un brick vengono replicati su altri nodi del cluster per garantire che, in caso di guasto di un nodo, i dati possano essere ancora accessibili dai client che utilizzano il cluster Gluster. In sintesi, un brick Gluster è una unità di spazio di archiviazione che viene utilizzata per fornire uno spazio di archiviazione distribuito su un cluster Gluster, contribuendo alla capacità totale di archiviazione del cluster e alla disponibilità dei dati.

Verifica delle porte di networking usate da Gluster

```
pietro@srv-gluster00:~$ netstat | grep gluster
tcp        0      0 srv-gluster00:49150  srv-gluster01:51736  ESTABLISHED
tcp        0      0 srv-gluster00:49147  srv-gluster02:24007  ESTABLISHED
tcp        0      0 localhost:49149      srv-gluster00:24007  ESTABLISHED
tcp        0      0 srv-gluster00:55651  localhost:49150      ESTABLISHED
tcp        0      0 srv-gluster00:49148  srv-gluster02:56795  ESTABLISHED
tcp        0      0 srv-gluster00:49149  srv-gluster01:51736  ESTABLISHED
tcp        0      0 srv-gluster00:ssh    _gateway:45363       ESTABLISHED
tcp        0      0 srv-gluster00:24007  localhost:49149      ESTABLISHED
tcp        0      0 srv-gluster00:55651  srv-gluster01:49149  ESTABLISHED
tcp        0      0 srv-gluster00:55651  localhost:49151      ESTABLISHED
tcp        0      0 srv-gluster00:24007  srv-gluster01:49150  ESTABLISHED
tcp        0      0 srv-gluster00:55651  172.16.10.100:49143  ESTABLISHED
tcp        0      0 srv-gluster00:24007  srv-gluster02:49150  ESTABLISHED
tcp        0      0 srv-gluster00:49151  srv-gluster01:24007  ESTABLISHED
tcp        0      0 srv-gluster00:55651  srv-gluster02:49149  ESTABLISHED
tcp        0      0 localhost:49151      srv-gluster00:55651  ESTABLISHED
tcp        0      0 srv-gluster00:55651  srv-gluster01:49151  ESTABLISHED
tcp        0      0 srv-gluster00:55651  srv-gluster02:49147  ESTABLISHED
tcp        0      0 srv-gluster00:24007  172.16.10.100:49150  ESTABLISHED
tcp        0      0 localhost:49150      srv-gluster00:55651  ESTABLISHED
tcp        0      0 srv-gluster00:ssh    _gateway:45359       ESTABLISHED
tcp        0      0 srv-gluster00:49146  srv-gluster02:56795  ESTABLISHED
unix 3      [ ]          STREAM     CONNECTED   33115
/var/run/gluster/09f8ebd3523ad603.socket
unix 3      [ ]          STREAM     CONNECTED   31598
/var/run/gluster/47002671169c2a56.socket
unix 3      [ ]          STREAM     CONNECTED   33291
/var/run/gluster/quotad.socket
unix 3      [ ]          STREAM     CONNECTED   31541
/var/run/gluster/8687366ca26a3310.socket
```

Accesso ai dati- Configurazione del client GlusterFS

Puoi accedere ai volumi di cluster in diversi modi. Puoi utilizzare il metodo Gluster Native Client per un'elevata concorrenza, prestazioni e failover trasparente nei client GNU/Linux. Puoi anche utilizzare NFS v3 per accedere ai volumi di cluster. Sono stati eseguiti test approfonditi sui client GNU/Linux e sull'implementazione NFS in altri sistemi operativi, come FreeBSD e Mac OS X, nonché Windows 7 (Professional e versioni successive) e Windows Server 2003. Altre implementazioni client NFS potrebbero funzionare con gluster Server NFS.

È possibile utilizzare CIFS per accedere ai volumi quando si utilizza Microsoft Windows e client SAMBA. Per questo metodo di accesso, i pacchetti Samba devono essere presenti sul lato client.

Client nativo di cluster

Il client nativo di Gluster è un client basato su FUSE in esecuzione nello spazio utente. Gluster Native Client è il metodo consigliato per l'accesso ai volumi quando sono richieste un'elevata concorrenza e prestazioni di scrittura elevate.

Questa sezione introduce Gluster Native Client e spiega come installare il software sui computer client.

Questa sezione descrive inoltre come montare i volumi sui client (sia manualmente che automaticamente) e come verificare che il volume sia stato montato correttamente.

Installazione del client nativo di cluster

Prima di iniziare l'installazione di Gluster Native Client, è necessario verificare che il modulo FUSE sia caricato sul client e abbia accesso ai moduli richiesti come segue:

Aggiungi il modulo kernel caricabile FUSE (LKM) al kernel Linux:

- `modprobe fuse`

Verificare che il modulo FUSE sia caricato:

```
2. # dmesg | grep -i fuse
    fuse init (API version 7.13)
```

Installazione su distribuzioni Red Hat Package Manager (RPM).

Per installare Gluster Native Client su sistemi basati su distribuzione RPM

Installa i prerequisiti richiesti sul client utilizzando il seguente comando:

- `sudo yum -y install openssh-server wget fuse fuse-libs openib libibverbs`

Assicurati che le porte TCP e UDP 24007 e 24008 siano aperte su tutti i server Gluster. Oltre a queste porte, è necessario aprire una porta per ogni mattone a partire dalla porta 49152 (anziché dalla 24009 in poi come con le versioni precedenti). Lo schema di assegnazione delle porte brick è ora conforme alle linee guida IANA. Ad esempio: se hai cinque brick, devi avere le porte da 49152 a 49156 aperte.

Da Gluster-10 in poi, i port brick verranno randomizzati. Una porta viene selezionata casualmente all'interno dell'intervallo da `base_port` a `max_port` come definito nel file `glusterd.vol` e quindi assegnata al brick. Ad esempio: se hai cinque brick, devi avere almeno 5 porte aperte all'interno dell'intervallo dato di **base_port e max_port**. Per ridurre il numero di porte aperte (per le migliori pratiche di sicurezza), è possibile abbassare il valore `max_port` nel file `glusterd.vol` e riavviare `glusterd` per renderlo effettivo.

È possibile utilizzare le seguenti catene con iptables:

```
sudo iptables -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 24007:24008 -j ACCEPT
```

```
sudo iptables -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 49152:49156 -j ACCEPT
```

Nota

Se hai già catene iptable, assicurati che le regole ACCEPT di cui sopra precedano le regole DROP. Ciò può essere ottenuto fornendo un numero di regola inferiore rispetto alla regola DROP.

Scarica gli ultimi file RPM `glusterfs`, `glusterfs-fuse` e `glusterfs-rdma` su ciascun client. Il pacchetto `glusterfs` contiene il client nativo di Gluster. Il pacchetto `glusterfs-fuse` contiene il traduttore FUSE necessario per il

montaggio sui sistemi client e i pacchetti glusterfs-rdma contengono il modulo RDMA dei verbi OpenFabrics per Infiniband.

È possibile scaricare il software dalla [pagina di download di GlusterFS](#).

Installa Gluster Native Client sul client.

Nota Le versioni del pacchetto elencate nell'esempio seguente potrebbero non essere l'ultima versione.

Fare riferimento alla pagina di download per assicurarsi di disporre dei pacchetti rilasciati di recente.

4. `sudo rpm -i glusterfs-3.8.5-1.x86_64`
5. `sudo rpm -i glusterfs-fuse-3.8.5-1.x86_64`
6. `sudo rpm -i glusterfs-rdma-3.8.5-1.x86_64`

Nota: il modulo RDMA è richiesto solo quando si utilizza Infiniband.

Installazione su distribuzioni basate su Debian

Per installare Gluster Native Client su distribuzioni basate su Debian

Installa OpenSSH Server su ciascun client utilizzando il seguente comando:

```
sudo apt-get install openssh-server vim wget
```

Scarica l'ultimo file .deb di GlusterFS e il checksum su ciascun client.

È possibile scaricare il software dalla [pagina di download di GlusterFS](#).

Per ogni file .deb, ottieni il checksum (usando il seguente comando) e confrontalo con il checksum per quel file nel file md5sum.

```
md5sum GlusterFS_DEB_file.deb
```

L'md5sum dei pacchetti è disponibile alla [pagina di download di GlusterFS](#)

Disinstalla GlusterFS v3.1 (o una versione precedente) dal client utilizzando il seguente comando:

```
sudo dpkg -r glusterfs
```

(Facoltativo) Esegui `$ sudo dpkg --purge glusterfs` per eliminare i file di configurazione.

Installa Gluster Native Client sul client utilizzando il seguente comando:

```
sudo dpkg -i GlusterFS_DEB_file
```

Per esempio:

```
sudo dpkg -i glusterfs-3.8.x.deb
```

Assicurati che le porte TCP e UDP 24007 e 24008 siano aperte su tutti i server Gluster. Oltre a queste porte, è necessario aprire una porta per ogni mattone a partire dalla porta 49152 (anziché dalla 24009 in poi come con le versioni precedenti). **Lo schema di assegnazione delle porte brick è ora conforme alle linee guida IANA.** Ad esempio: se hai cinque brick, devi avere le porte da 49152 a 49156 aperte.

Da Gluster-10 in poi, i port brick verranno randomizzati. Una porta viene selezionata casualmente all'interno dell'intervallo da **base_port a max_port** come definito **nel file glusterd.vol** e quindi assegnata al brick. Ad esempio: se hai cinque brick, devi avere almeno 5 porte aperte all'interno dell'intervallo dato di base_port e max_port. Per ridurre il numero di porte aperte (**per le migliori pratiche di sicurezza**), è possibile abbassare il valore max_port nel file glusterd.vol e riavviare glusterd per renderlo effettivo.

È possibile utilizzare le seguenti catene con iptables:

6. `sudo iptables -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 24007:24008 -j ACCEPT`
- 7.
8. `sudo iptables -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 49152:49156 -j ACCEPT`

Nota

Se hai già catene iptable, assicurati che le regole ACCEPT di cui sopra precedano le regole DROP. Ciò può essere ottenuto fornendo un numero di regola inferiore rispetto alla regola DROP.

Montaggio dei volumi Gluster

Dopo aver installato Gluster Native Client, è necessario montare i volumi Gluster per accedere ai dati. Ci sono due metodi che puoi scegliere:

1. Montaggio manuale dei volumi
2. Volumi di montaggio automatico

Nota

I nomi dei server selezionati durante la creazione dei volumi devono essere risolvibili nel computer client. È possibile utilizzare le voci `/etc/hosts` appropriate o il server DNS per risolvere i nomi dei server in indirizzi IP.

Montaggio manuale dei volumi

Per montare un volume, utilizzare il seguente comando:

- `mount -t glusterfs HOSTNAME-OR-IPADDRESS:/VOLNAME MOUNTDIR`

Per esempio:

```
mount -t glusterfs server1:/test-volume /mnt/glusterfs
```

Nota

Il server specificato nel comando `mount` viene utilizzato solo per recuperare il volfile di configurazione del cluster che descrive il nome del volume. Successivamente, il client comunicherà direttamente con i server menzionati nel volfile (che potrebbe anche non includere quello utilizzato per il mount).

Se viene visualizzato un messaggio di utilizzo come "Utilizzo: mount.glusterfs", `mount` di solito richiede la creazione di una directory da utilizzare come punto di montaggio. Esegui `"mkdir /mnt/glusterfs"` prima di tentare di eseguire il comando `mount` sopra elencato.

Opzioni di montaggio

È possibile specificare le seguenti opzioni quando si utilizza il `mount -t glusterfs` comando. Tieni presente che devi separare tutte le opzioni con le virgole.

```
backupvolfile-server=server-name
volfile-max-fetch-attempts=number of attempts
log-level=loglevel
log-file=logfile
transport=transport-type
direct-io-mode=[enable|disable]
use-readdirp=[yes|no]
```

Per esempio:

```
mount -t glusterfs -o backupvolfile-server=volfile_server2,use-readdirp=no,volfile-max-fetch-attempts=2,log-level=WARNING,log-file=/var/log/gluster.log server1:/test-volume /mnt/glusterfs
```

Se `backupvolfile-server` l'opzione viene aggiunta durante il montaggio del client fuse, quando il primo server volfile fallisce, il server specificato nell'opzione `backupvolfile-server` viene utilizzato come server volfile per montare il client.

In `volfile-max-fetch-attempts=X` opzione, specificare il numero di tentativi per recuperare i file del volume durante il montaggio di un volume. Questa opzione è utile quando monti un server con più indirizzi IP o quando il DNS round-robin è configurato per il nome del server.

Se `use-readdirp` è impostato su ON, forza l'uso della modalità `readdirp` nel modulo fuse kernel

Montaggio automatico dei volumi Gluster

Puoi configurare il tuo sistema per montare automaticamente il volume Gluster ogni volta che il tuo sistema viene avviato. Il server specificato nel comando `mount` viene utilizzato solo per recuperare il volfile di configurazione del cluster che descrive il nome del volume. Successivamente, il client comunicherà direttamente con i server menzionati nel volfile (che potrebbe anche non includere quello utilizzato per il mount).

Per montare un volume, modifica il file `/etc/fstab` e aggiungi la seguente riga:

```
HOSTNAME-OR-IPADDRESS:/VOLNAME MOUNTDIR glusterfs defaults,_netdev 0 0
```

Per esempio:

```
server1:/test-volume /mnt/glusterfs glusterfs defaults,_netdev 0 0
```

Opzioni di montaggio

È possibile specificare le seguenti opzioni durante l'aggiornamento del file `/etc/fstab`. Tieni presente che devi separare tutte le opzioni con le virgole.

```
log-level=loglevel  
log-file=logfile  
transport=transport-type  
direct-io-mode=[enable|disable]  
use-readdirp=no
```

Per esempio:

```
HOSTNAME-OR-IPADDRESS:/VOLNAME MOUNTDIR glusterfs defaults,_netdev,log-  
level=WARNING,log-file=/var/log/gluster.log 0 0
```

Testare i volumi montati

Per testare i volumi montati

Usa il seguente comando:

```
# mount
```

Se il volume cluster è stato montato correttamente, l'output del comando `mount` sul client sarà simile a questo esempio:

```
server1:/test-volume on /mnt/glusterfs type fuse.glusterfs  
(rw,allow_other,default_permissions,max_read=131072
```

Usa il seguente comando:

```
# df
```

L'output del comando `df` sul client visualizzerà lo spazio di archiviazione aggregato da tutti i mattoni in un volume simile a questo esempio:

```
# df -h /mnt/glusterfs  
Filesystem          Size Used Avail Use% Mounted on  
server1:/test-volume 28T 22T 5.4T 82% /mnt/glusterfs
```

Passare alla directory ed elencare i contenuti inserendo quanto segue:

```
`# cd MOUNTDIR `  
`# ls`
```

Per esempio,

```
`# cd /mnt/glusterfs `  
`# ls`
```

Protocollo NFS (Network File System)

È possibile utilizzare NFS v3 per accedere ai volumi cluster. Sono stati eseguiti test approfonditi sui client GNU/Linux e l'implementazione NFS in altri sistemi operativi, come FreeBSD e Mac OS X, nonché Windows 7 (Professional e versioni successive), Windows Server 2003 e altri, potrebbero funzionare con gluster NFS implementazione del server.

GlusterFS ora include Network Lock Manager (NLM) v4. NLM consente alle applicazioni sui client NFSv3 di eseguire il blocco dei record sui file sul server NFS. Viene avviato automaticamente ogni volta che viene eseguito il server NFS. È necessario installare il pacchetto `nfs-common` sia sui server che sui client (solo per la distribuzione basata su Debian). Questa sezione descrive come utilizzare NFS per montare i volumi Gluster (sia manualmente che automaticamente) e come verificare che il volume sia stato montato correttamente.

Utilizzo di NFS per montare i volumi

Puoi utilizzare uno dei seguenti metodi per montare i volumi Gluster:

1. Montaggio manuale dei volumi tramite NFS
2. Montaggio automatico dei volumi tramite NFS

Prerequisito : installare il pacchetto `nfs-common` sia sui server che sui client (solo per la distribuzione basata su Debian), utilizzando il seguente comando:

```
sudo aptitude install nfs-common
```

Montaggio manuale dei volumi tramite NFS

Per montare un volume, utilizzare il seguente comando:

- `mount -t nfs -o vers=3 HOSTNAME-OR-IPADDRESS:/VOLNAME MOUNTDIR`

Per esempio:

```
mount -t nfs -o vers=3 server1:/test-volume /mnt/glusterfs
```

Nota

Il server NFS di cluster non supporta UDP. Se per impostazione predefinita il client NFS in uso si connette tramite UDP, viene visualizzato il seguente messaggio:

```
requested NFS version or transport protocol is not supported.
```

Per connettersi tramite TCP

Aggiungere la seguente opzione al comando `mount`:

```
-o mountproto=tcp
```

Per esempio:

```
mount -o mountproto=tcp -t nfs server1:/test-volume /mnt/glusterfs
```

Per montare il server Gluster NFS da un client Solaris

Usa il seguente comando:

- `mount -o proto=tcp,vers=3 nfs://HOSTNAME-OR-IPADDRESS:38467/VOLNAME MOUNTDIR`

Per esempio:

```
mount -o proto=tcp,vers=3 nfs://server1:38467/test-volume /mnt/glusterfs
```

Montaggio automatico dei volumi tramite NFS

Puoi configurare il tuo sistema per montare automaticamente i volumi Gluster utilizzando NFS ogni volta che il sistema viene avviato.

Per montare automaticamente un volume Gluster utilizzando NFS

Per montare un volume, modifica il file `/etc/fstab` e aggiungi la seguente riga:

```
HOSTNAME-OR-IPADDRESS:/VOLNAME MOUNTDIR nfs defaults,_netdev,vers=3 0 0
```

Per esempio,

```
server1:/test-volume /mnt/glusterfs nfs defaults,_netdev,vers=3 0 0
```

Nota

Il server NFS di cluster non supporta UDP. Se per impostazione predefinita il client NFS in uso si connette tramite UDP, viene visualizzato il seguente messaggio:

```
requested NFS version or transport protocol is not supported.
```

Per connettersi tramite TCP

Aggiungere la seguente voce nel file `/etc/fstab` :

- `HOSTNAME-OR-IPADDRESS:/VOLNAME MOUNTDIR nfs defaults,_netdev,mountproto=tcp 0 0`

Per esempio,

```
server1:/test-volume /mnt/glusterfs nfs defaults,_netdev,mountproto=tcp 0 0
```

Per montare automaticamente i montaggi NFS

Gluster supporta il metodo standard *nix di montaggio automatico dei montaggi NFS. Aggiorna `/etc/auto.master` e `/etc/auto.misc` e riavvia il servizio autofs. Successivamente, ogni volta che un utente o un processo tenta di accedere alla directory, verrà montato in background.

Testare i volumi montati utilizzando NFS

Puoi confermare che le directory Gluster vengono montate correttamente.

Utilizzare il comando `mount` immettendo quanto segue:

```
# mount
```

Ad esempio, l'output del comando `mount` sul client visualizzerà una voce come la seguente:

```
server1:/test-volume on /mnt/glusterfs type nfs (rw,vers=3,addr=server1)
```

Utilizzare il comando `df` inserendo quanto segue:

```
# df
```

Ad esempio, l'output del comando `df` sul client visualizzerà lo spazio di archiviazione aggregato da tutti i mattoni in un volume.

```
# df -h /mnt/glusterfs
Filesystem                Size Used Avail Use% Mounted on
server1:/test-volume      28T  22T  5.4T  82% /mnt/glusterfs
```

Passare alla directory ed elencare i contenuti inserendo quanto segue:

```
# cd MOUNTDIR
```

```
# ls
```

Protocollo MS Windows CIFS e SAMBA

È possibile utilizzare CIFS per accedere ai volumi quando si utilizza Microsoft Windows e client SAMBA. Per questo metodo di accesso, i pacchetti Samba devono essere presenti sul lato client. È possibile esportare il punto di montaggio glusterfs come esportazione samba e quindi montarlo utilizzando il protocollo CIFS. Questa sezione descrive come montare le condivisioni CIFS su client basati su Microsoft Windows (sia manualmente che automaticamente) e come verificare che il volume sia stato montato correttamente.

Nota

L'accesso a CIFS utilizzando il Finder di Mac OS X non è supportato, tuttavia, puoi utilizzare la riga di comando di Mac OS X per accedere ai volumi Gluster utilizzando CIFS.

Utilizzo di CIFS per montare i volumi

Puoi utilizzare uno dei seguenti metodi per montare i volumi Gluster:

1. Esportazione di volumi Gluster tramite Samba
2. Montaggio manuale dei volumi tramite CIFS
3. Montaggio automatico dei volumi tramite CIFS

Puoi anche utilizzare Samba per esportare i volumi Gluster tramite il protocollo CIFS.

Esportazione di volumi Gluster tramite Samba

Ti consigliamo di utilizzare Samba per esportare i volumi Gluster tramite il protocollo CIFS.

Per esportare volumi tramite protocollo CIFS

1. Montare un volume Gluster.
2. Imposta la configurazione di Samba per esportare il punto di montaggio del volume Gluster.

Ad esempio, se un volume Gluster è montato su `/mnt/gluster`, è necessario modificare il file `smb.conf` per abilitare l'esportazione tramite CIFS. Apri il file `smb.conf` in un editor e aggiungi le seguenti righe per una semplice configurazione:

```
[glustertest]
comment = For testing a Gluster volume exported through CIFS
path = /mnt/glusterfs
read only = no
guest ok = yes
```

Salva le modifiche e avvia il servizio `smb` utilizzando gli script `init` del tuo sistema (`/etc/init.d/smb [re]start`). I passaggi sopra sono necessari per eseguire più montaggi. Se vuoi solo montare samba, allora nel tuo `smb.conf` devi aggiungere

```
kernel share modes = no
kernel oplocks = no
map archive = no
map hidden = no
map read only = no
map system = no
store dos attributes = yes
```

Nota

Per poter eseguire il montaggio da qualsiasi server nel pool di archiviazione attendibile, è necessario ripetere questi passaggi su ciascun nodo Gluster. Per configurazioni più avanzate, vedere la documentazione di Samba.

Montaggio manuale dei volumi tramite CIFS

È possibile montare manualmente i volumi Gluster utilizzando CIFS su computer client basati su Microsoft Windows.

1. Utilizzando Windows Explorer, scegli **Strumenti > Connetti unità di rete...** dal menu. Viene visualizzata la finestra **Connetti unità di rete**.
2. Scegli la lettera dell'unità utilizzando l'elenco a discesa **Unità**.
3. Fare clic su **Sfogliala**, selezionare il volume da mappare all'unità di rete e fare clic su **OK**.
4. Fare clic su **Fine**.

L'unità di rete (associata al volume) viene visualizzata nella finestra Computer.

In alternativa, per montare manualmente un volume Gluster utilizzando CIFS, vai su **Start > Esegui** e immetti Percorso di rete manualmente.

Montaggio automatico dei volumi tramite CIFS

Puoi configurare il tuo sistema per montare automaticamente i volumi Gluster utilizzando CIFS su client basati su Microsoft Windows ogni volta che il sistema viene avviato.

L'unità di rete (associata al volume) viene visualizzata nella finestra Computer e viene riconnessa a ogni avvio del sistema.

1. Utilizzando Windows Explorer, scegli **Strumenti > Connetti unità di rete...** dal menu. Viene visualizzata la finestra **Connetti unità di rete** .
2. Scegli la lettera dell'unità utilizzando l'elenco a discesa **Unità** .
3. Fare clic su **Sfoglia** , selezionare il volume da mappare all'unità di rete e fare clic su **OK** .
4. Fare clic sulla casella di controllo **Riconnetti all'accesso** .
5. Fare clic su **Fine**.

Testare i volumi montati utilizzando CIFS

Puoi verificare che le directory di Gluster vengano montate correttamente accedendo alla directory utilizzando Esplora risorse.

NFS-Ganesha su GlusterFS

NFS-Ganesha è un file server in spazio utente (user land) per il protocollo NFS con supporto per NFSv3, v4, v4.1, pNFS. Fornisce un File System Abstraction Layer (FSAL) compatibile con FUSE per consentire agli sviluppatori di file system di collegare il proprio meccanismo di archiviazione e accedervi da qualsiasi client NFS. NFS-Ganesha può accedere ai filesystem FUSE direttamente tramite il suo FSAL senza copiare alcun dato da o verso il kernel, migliorando così potenzialmente i tempi di risposta.

Configurazione

Da completare...

Installazione

Da completare...

Mount

Da completare...

Test

Da completare...

ⁱ Use **default** settings. Equivalent to rw, suid, dev, exec, auto, nouser, async.

_netdev : this is a network device, mount it after bringing up the network. Only valid with fstype nfs

Trattandosi di un volume di rete, il check del file system non va mai fatto.

Di seguito le info di dettaglio di fsck:

Pass (fsck order)

Fsck order is to tell fsck what order to check the file systems, if set to "0" file system is ignored.

Often a source of confusion, there are only 3 options :

0 == do not check.

1 == check this partition first.

2 == check this partition(s) next

In practice, use "1" for your root partition, / and 2 for the rest. All partitions marked with a "2" are checked in sequence and you do not need to specify an order.

Use "0" to disable checking the file system at boot or for network shares.