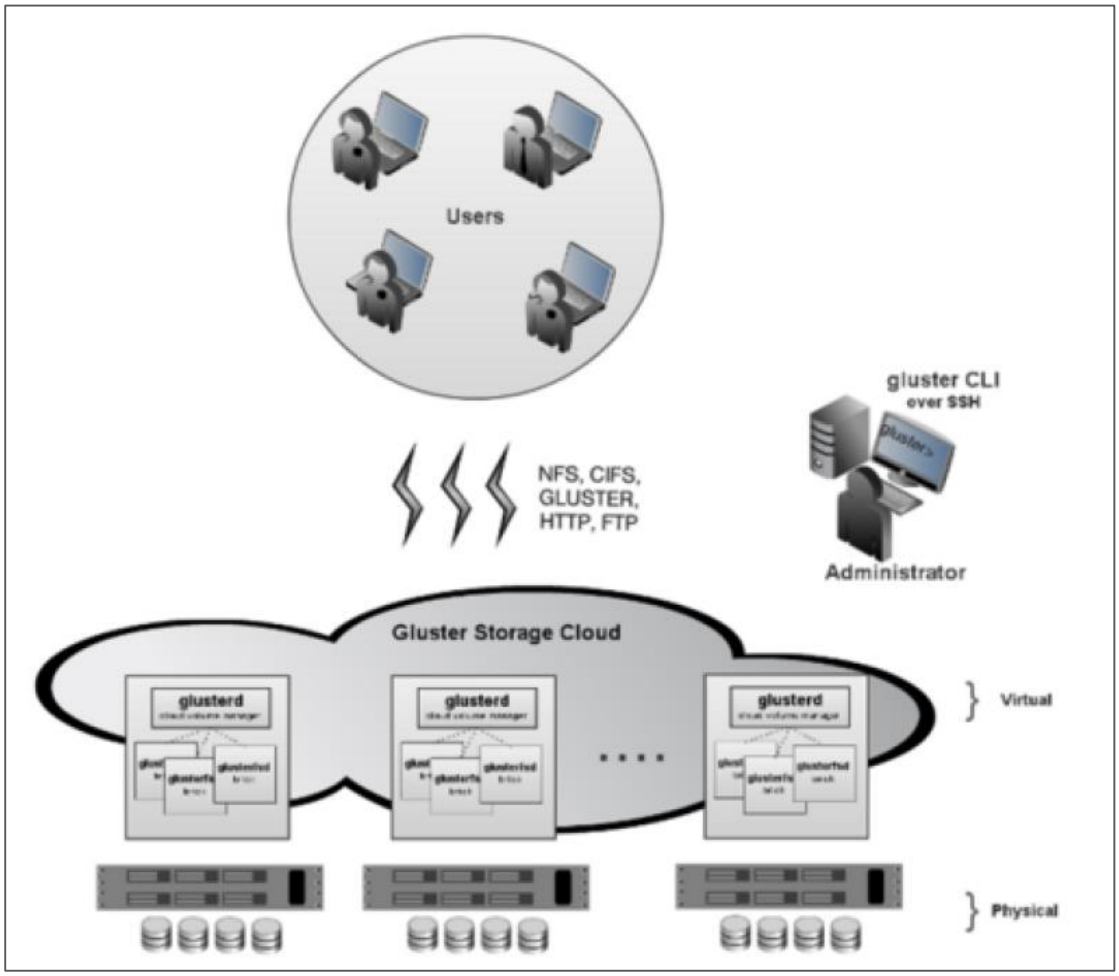
Preface

Gluster 是一個分散式 Scale-Out 檔案系統，依據客戶端儲存體空間的需求可快速供應(Provisioning)你額外的儲存空間。它是一套開放原始碼軟體，可群聚多個檔案系統提供數以 PB 計的儲存資源(據官方網站說明，實際上可達到 72 brontobytes)及上千個用戶端作資源存取。GlusterFS 在傳統的系統環境及低成本的情況下，可彈性的整合實體、虛擬及雲端儲存資源以實現高可用性及高效能的企業儲存資源。除此之外，還可透過高速頻寬的 Infiniband RDMA 或 TCP/IP 協定在多個叢集儲存節點上互連，並在單一全域命名空間上整合多個節點上的磁碟、記憶體資源及資料管理。

綜合 Gluster 特性，分列整理如下:

I. Scalability and Performance
II. High Availability
III. Global Namespace
IV. Elastic Hash Algorithm
V. Elastic Volume Manager
VI. Gluster Console Manager
VII. Standards-based



LAB Environment		
OS: CentOS 6.3 (2.6.32-279.el6.x86_64)		
Node 1 (Gluster1)	eth0	172.20.144.83 (Mgmt)
	eth1	1.1.1.1 (Mirror & Heartbeat)
	Disk (Brick)	/dev/sda2
Node 2 (Gluster2)	eth0	172.20.144.84 (Mgmt)
	eth1	1.1.1.2 (Mirror & Heartbeat)
	Disk (Brick)	/dev/sda2
Virtual IP	eth0:0	172.20.144.80
Gluster Volume Name	gv0	

1. 安裝 Gluster (YUM 自動安裝)

I. 於 CentOS 作業系統安裝時，選擇 Basic Server 安裝。

II. 使用 wget 工具，下載 Gluster YUM Repository。若連結失效請自行至該網站找最新連結。

III. 若無法使用 YUM 來自動安裝，可參考下一則手動安裝參考。

```
wget -P /etc/yum.repos.d
http://download.gluster.org/pub/gluster/glusterfs/LATEST/EPEL.repo/glusterfs-epel.repo
```

IV. 安裝 Gluster

```
yum -y install glusterfs{-fuse,-server}
```

2. 安裝 Gluster (無法連至 Internet，需手動安裝)

I. 於 CentOS 安裝時，選擇 Basic Server 安裝。

II. 準備 Gluster 安裝套件

```
glusterfs-3.3.1-1.el6.x86_64.rpm
glusterfs-server-3.3.1-1.el7.x86_64.rpm
glusterfs-fuse-3.3.1-1.el7.x86_64.rpm
```

III. 安裝 Gluster

```
# rpm -ivh glusterfs-3.3.1-1.el6.x86_64.rpm glusterfs-server-3.3.1-1.el7.x86_64.rpm
glusterfs-fuse-3.3.1-1.el7.x86_64.rpm
```

3. Nodes 主機基礎設定 (兩台 Nodes 都需設定)

I. 命名主機名稱(Hostname)

Node 1: Gluster1
Node 2: Gluster2

II. 關閉 iptables 及 SELinux 服務

III. 修改 /etc/hosts，加入兩台主機名稱及 IP。

```
[root@Gluster1 ~]# vim /etc/hosts
1.1.1.1    Gluster1
1.1.1.2    Gluster2
```

IV. IP 配置，包含管理 IP(172.20.144.x)及 Mirror IP(1.1.1.x)

V. 配置 Bricks(Mirror Disk)，本例使用裝置 /dev/sda2。

```
[root@Gluster1 ~]# fdisk /dev/sda
```

VI. 格式化為 XFS 檔案系統，並設定開機自動掛載。(需先安裝 xfsdump 套件)

```
[root@Gluster1 ~]# yum -y install xfs
[root@Gluster1 ~]# mkdir -p /Brick1        //建立/dev/sda2 掛載目錄
[root@Gluster1 ~]# mkfs.xfs -i size=512 /dev/sda2        //格式化為 XFS 檔案系統
[root@Gluster1 ~]# vim /etc/fstab        //設定開機自動掛載
/dev/sda2 /Brick1 xfs defaults 1 2
[root@Gluster1 ~]# mount -a && mount
```

4. 使用 Gluster Console Manager – 命令列工具

I. Gluster Console Manager 是一個可下達指令的工具程式，它簡化了設定及儲存環境的管理，並可在 1 台節點同步下達指令給所有的節點。透過此工具能夠建立新的 Volume、啟動及停止 Volume 的服務，亦可對線上的 Volume 進行動態的 bricks 新增、刪除修改。Gluster Console Manager 在 SSH Console 下分兩種操作模式，一是直接帶指令執行，另一方法則是進入互動模式來下達指令。

II. 範例 1: 直接執行指令

```
# gluster peer command
```

For example:
gluster peer status

III. 範例 2: 進入互動模式

gluster
You can execute gluster commands from the Console Manager prompt:
gluster> *command*
For example:
gluster
gluster > peer status

5. 設定信任儲存池(Trusted Storage Pools)

I. 在開始配置 Gluster volume 之前，必須要先建立所有節點(儲存伺服器)間溝通的信任儲存池。在開始啟動第一台節點時，預設是只能看到本機，必須透過信任儲存池將其他可信任的節點加入進來，成為一個大型的儲存池。

II. 將本機(Gluster1)以外節點(Gluster2)加入信任儲存池

[root@Gluster1 ~]# gluster peer probe gluster2
Probe successful

III. 驗證節點狀態

[root@Gluster1 ~]# gluster peer status
Number of Peers: 1

Hostname: Gluster2
Uuid: ad9d26e6-50ac-4273-a2f1-186e6d910ceb
State: Peer in Cluster (Connected)

IV. 從信任儲存池中移除節點

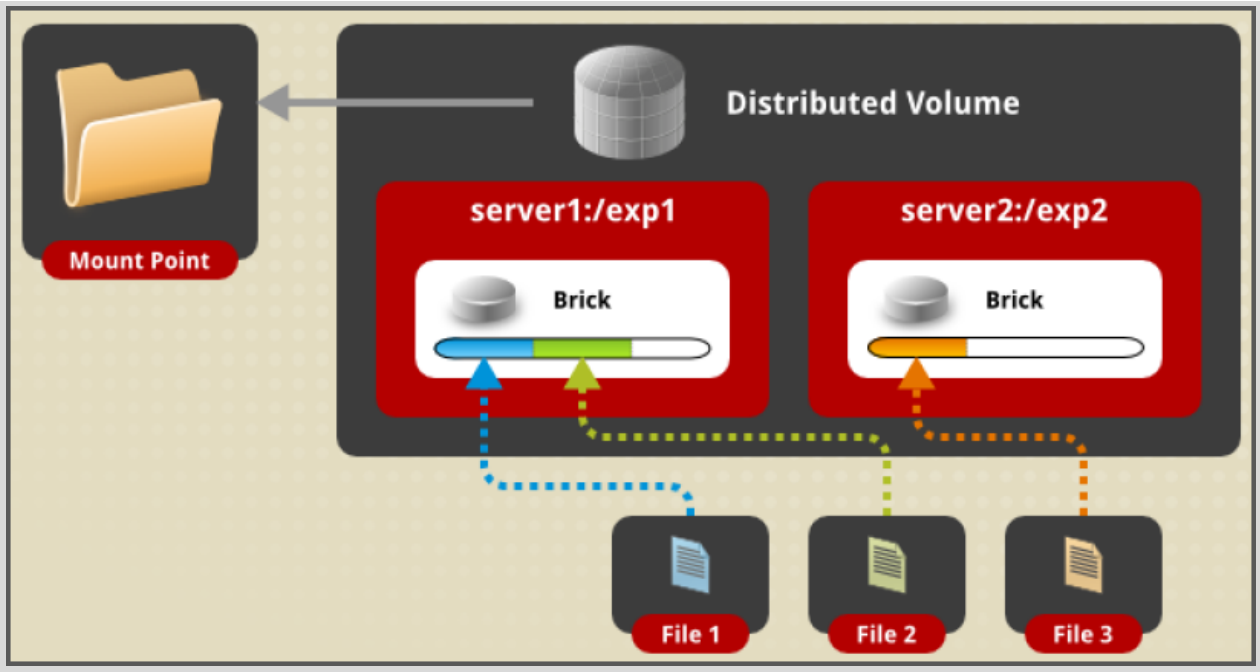
[root@Gluster1 ~]# gluster peer detach gluster2
Detach successful

6. 設置 Gluster Volume (只需在 Node1 執行)

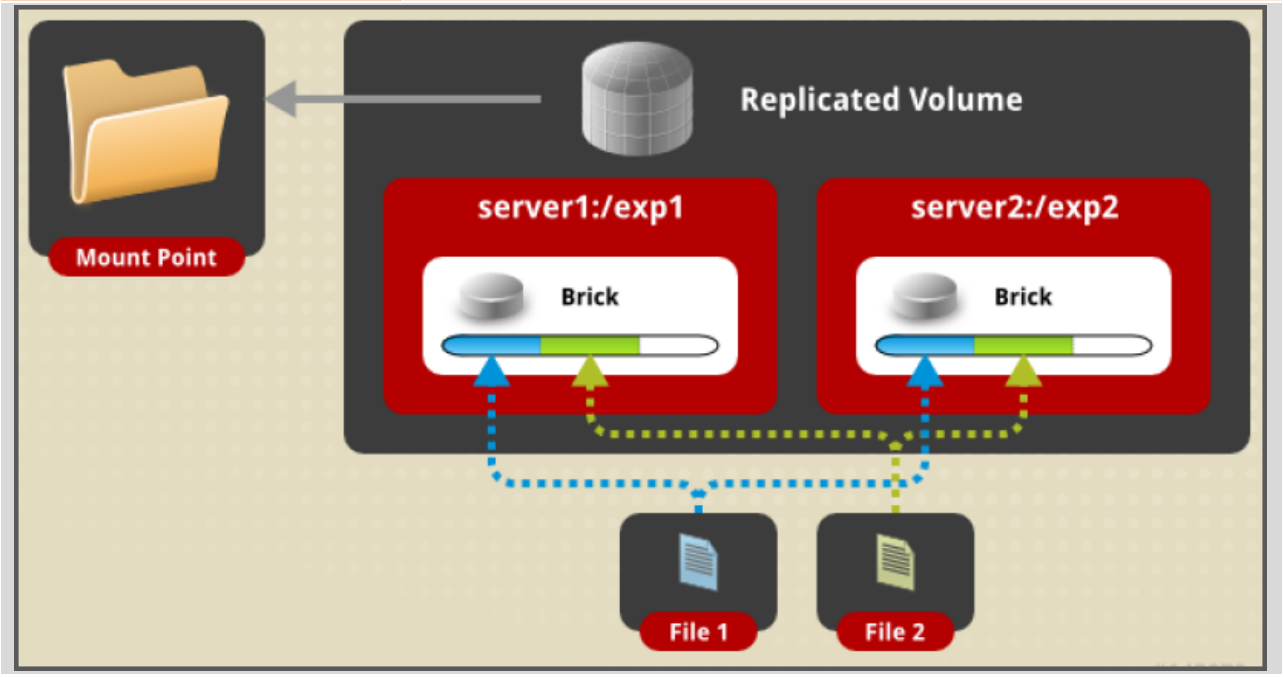
I. Gluster Volume 在這邊指的是各個 Node 主機端磁碟(Bricks)邏輯上的集合，可以想像將數顆實體的磁碟虛擬成一顆可識別的邏輯磁碟。

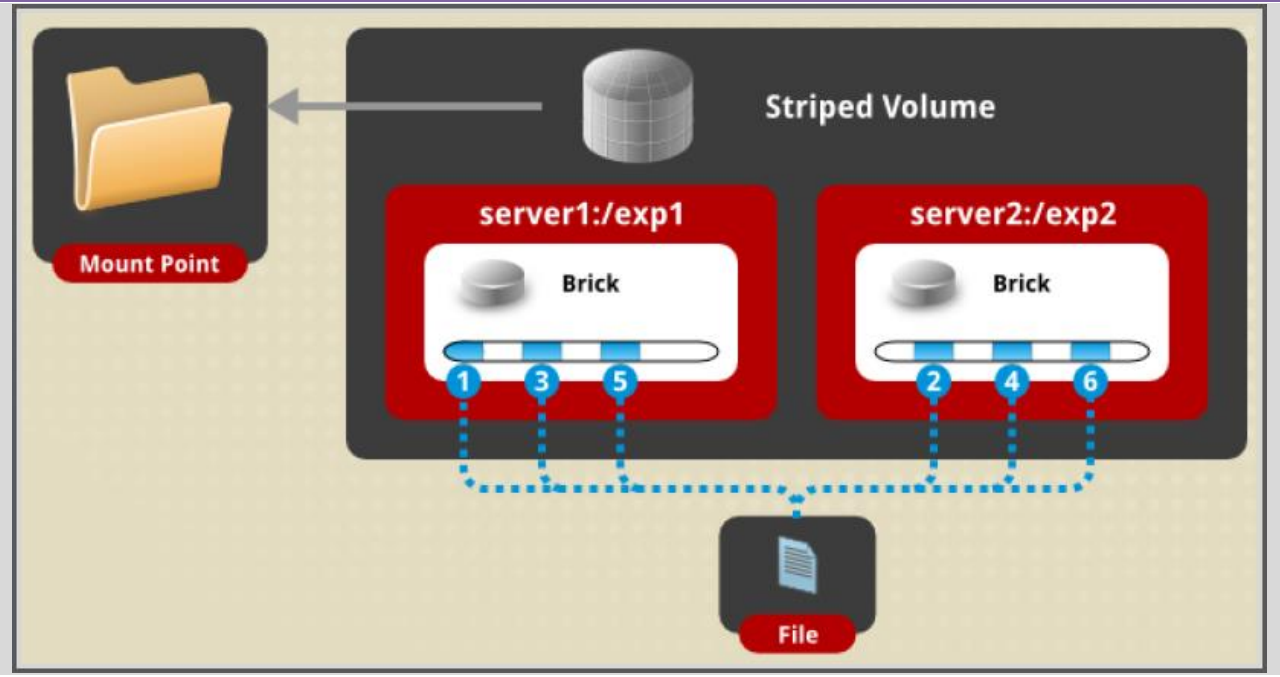
II. Gluster Volume 於儲存環境上，依照需求提供七種不同型態的儲存方法。

Volume Type	Description
1. Distributed	使用 Distributed，檔案將會隨機被儲存至不同的 Node 及 Bricks 上。需注意的是，當其中一台 Node 或磁碟發生故障，將會導致資料遺失問題。 範例: # gluster volume create NEW-VOLNAME [transport [tcp | rdma | tcp,rdma]] NEW-BRICK...

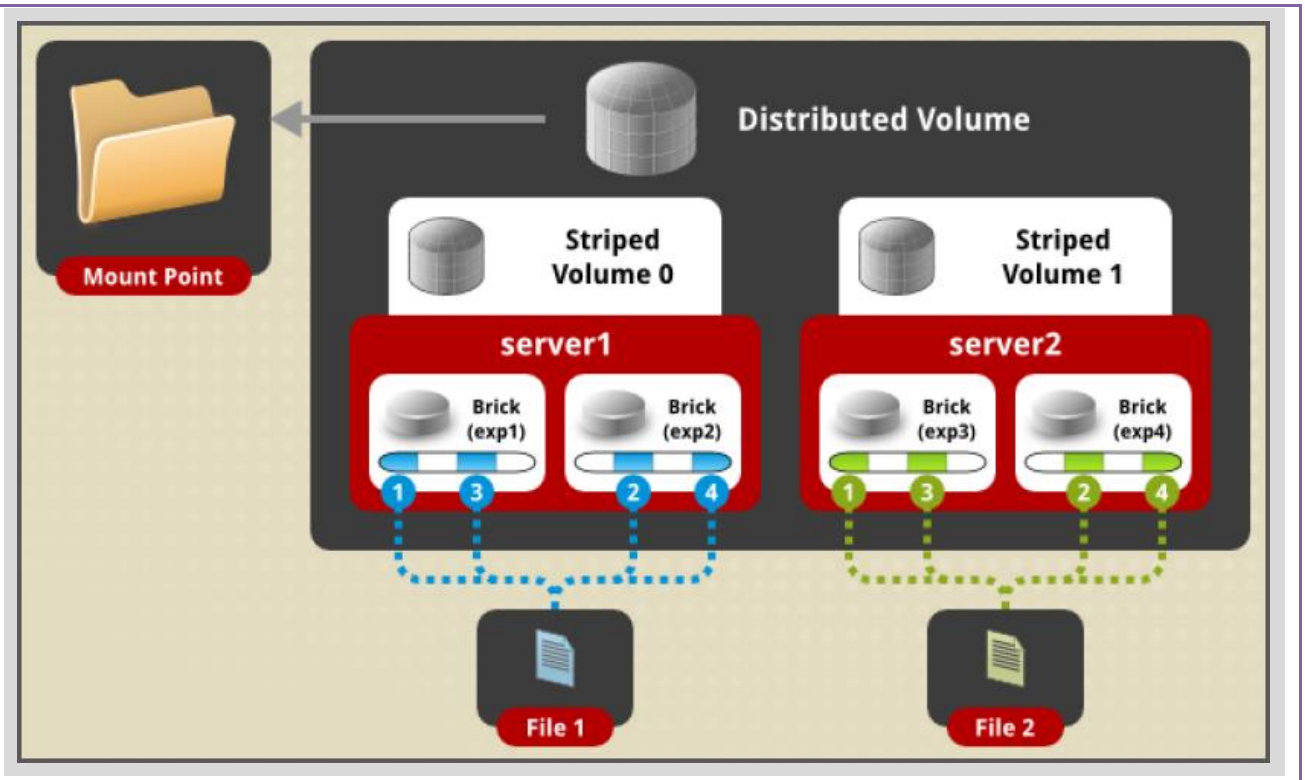


2. Replicated	使用 Replicated，檔案將會同時複寫多份至 Volume 下所有的 Bricks。此方法適合設計在需高可用性及高可靠性的企業上。

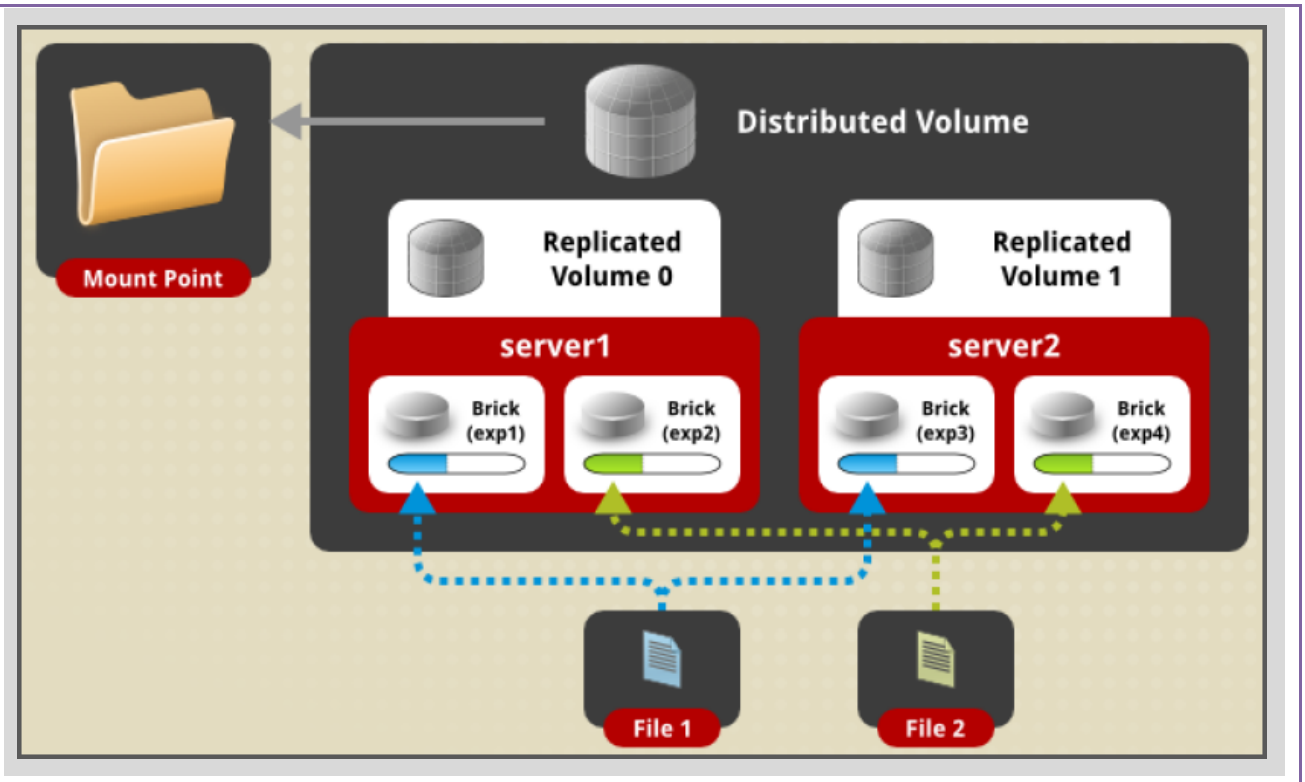
	需注意的是，所規劃的 **Replicated** 數量需等於目前配置的 **Bricks** 數量。 **範例:** # gluster volume create NEW-VOLNAME [replica COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK... # gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp2
	
3. Striped	使用 **Striped**，檔案將會被拆成數份儲存至被分配的 **Bricks** 上。需注意的是，所規劃的 **Striped** 數量需等於目前配置的 **Bricks** 數量。 **範例:** # gluster volume create NEW-VOLNAME [stripe COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK... # gluster volume create test-volume stripe 2 transport tcp server1:/exp1 server2:/exp2



4. Distributed Striped	**使用 Distributed Striped，檔案將會隨機被儲存至任一個 Node，且被拆成數位儲存至被分配的 Bricks 上。Bricks 數量必須要為 Stripe count 的倍數，例如 2 個 Nodes，Bricks 數量就要為 4 個。** **範例:** # gluster volume create NEW-VOLNAME [stripe COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK... # gluster volume create test-volume stripe 4 transport tcp server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4 server5:/exp5 server6:/exp6 server7:/exp7 server8:/exp8



5. Distributed Replicated	**使用 Distributed Replicated，檔案將會被複制至數個 Nodes(Replicated volume)，且被分散儲存在其一 Brick 上。** **Bricks 數量必須要為 Replicate count 的倍數，例如 2 個 Nodes，Bricks 數量就要為 4 個。** **範例:** # gluster volume create NEW-VOLNAME [replica COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK... # gluster volume create test-volume replica 2 transport tcp server1:/exp1 server2:/exp3 server1:/exp2 server2:/exp4



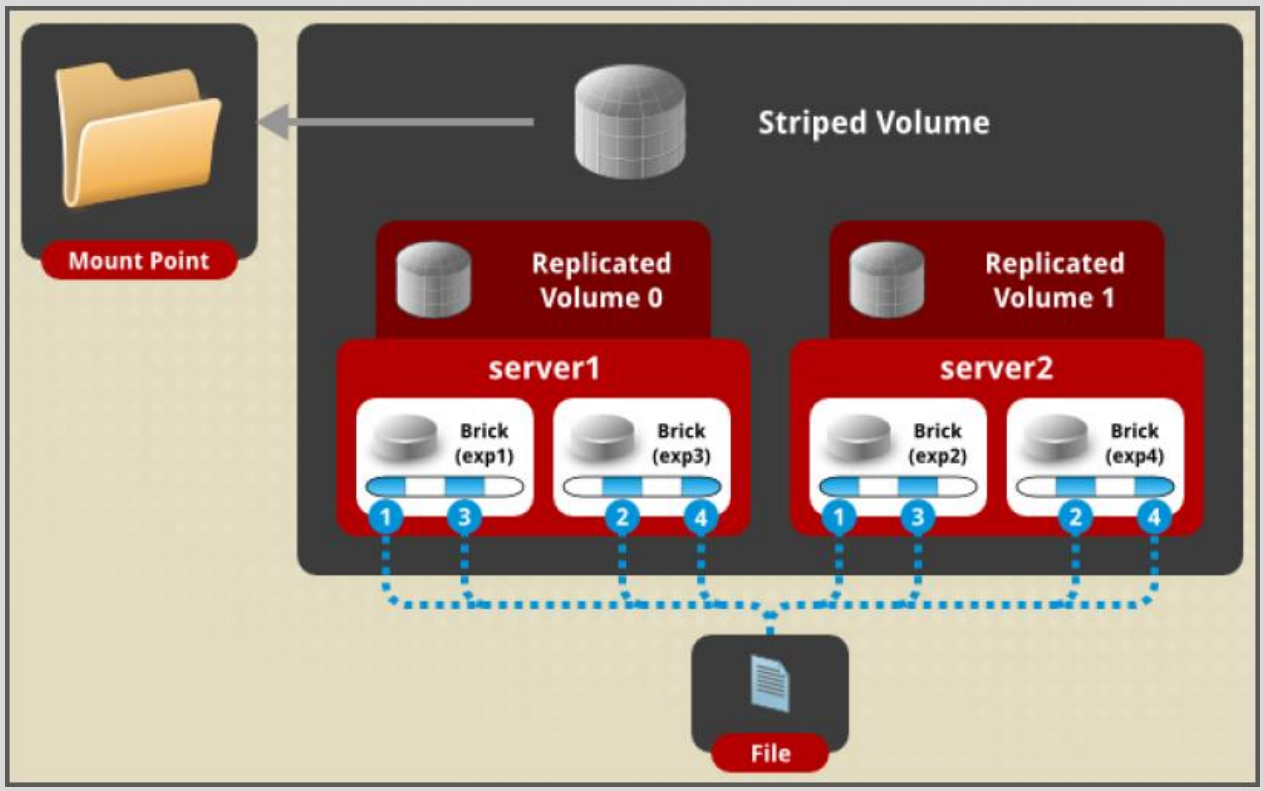
6. Distributed Striped Replicated	**使用 Distributed Striped Replicated，檔案將會被拆成(Striped)數個，且分散(Distributed)儲存至多個複制(Replicated)的 Bricks 上。Bricks 數量必須要為 Stripe count 及 Replicate count 的倍數。** **範例:** # gluster volume create NEW-VOLNAME [stripe COUNT] [replica COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK... # gluster volume create test-volume stripe 2 replica 2 transport tcp server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4 server5:/exp5 server6:/exp6 server7:/exp7 server8:/exp8
7. Striped Replicated	**使用 Striped Replicated，檔案將會被複制到多個 Nodes，且被拆散儲存至各 Node 內不同的 Bricks 上。Bricks 數量必須要為**

Stripe count 及 Replicate count 的倍數。

範例:

gluster volume create NEW-VOLNAME [stripe COUNT] [replica COUNT] [transport [tcp | rdma | tcp,rdma]] NEW-BRICK...

gluster volume create test-volume stripe 3 replica 2 transport tcp server1:/exp1 server2:/exp2 server3:/exp3 server4:/exp4 server5:/exp5 server6:/exp6



III. 本範例採用 replicate 方法來作 2 台 Nodes 間檔案的複制及保護。

[root@Gluster1 ~]# service glusterd start //啟動 Gluster Daemon

Starting glusterd: *[OK]*

[root@Gluster1 ~]# gluster volume create gv0 replica 2 transport tcp Gluster1:/Brick1 Gluster2:/Brick2 //建立 replicated volume

Creation of test-volume has been successful

Please start the volume to access data.

[root@Gluster1 ~]# gluster volume start gv0 //啟動 gv0 volume

Starting volume gv0 has been successful

[root@Gluster1 ~]# gluster volume info //確認 volume 資訊

Volume Name: gv0
Type: Replicate
Volume ID: 8f9331d8-e0ab-49f4-9160-b35686337e0f
Status: Started
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: gluster1:/Brick1
Brick2: gluster2:/Brick1

IV. 手動掛載 gluster 檔案系統(glusterfs) (2 個 Nodes 都需設定)

```
[root@Gluster1 ~]# mkdir -p /GShare
[root@Gluster1 ~]# mount -t glusterfs gluster1:/gv0 /GShare
[root@Gluster1 ~]# mount
/dev/sda2 on /Brick1 type xfs (rw)
Gluster1:/gv0 on /GShare type fuse.glusterfs
(rw,default_permissions,allow_other,max_read=131072)
```

V. 設定開機自動掛載 gluster 檔案系統(glusterfs) (2 個 Nodes 都需設定)

[root@Gluster1 ~]# vim /etc/fstab //加入下列一行

```
Gluster1:/gv0           /GShare          glusterfs defaults,_netdev      0 0
```

7. 測試 Gluster volume 資料寫入

I. 透過 for 迴圈連續寫入 100 筆檔案到 gluster volume

```
[root@Gluster1 ~]# for i in `seq -w 1 100`; do cp -rp /var/log/messages* /GShare/test-$i; done
```

II. 確認所有 Nodes 檔案數量是否一致

```
[root@Gluster1 ~]# ls -al /GShare | wc -l ; ssh Gluster2 ls -al /GShare | wc -l
```

8. Heartbeat (兩台 Nodes 都要設定)

I. 本範例將透過 Heartbeat 套件來提供一組對外 Virtual IP，且 2 台 Nodes 會定時互相偵測對方是否存活，並由 Heartbeat 服務來指派哪一台為 Primary 及 Secondary。預設情況下，外部檔案只會經由 Virtual IP 傳給 Primary Node，則 Secondary Node 只會被動接收 Replicated 的檔案。當 Primary Node 網卡異常或設備故障時，Heartbeat 服務會立即判斷並將 Virtual IP 重新指向給 Secondary Node。

II. 安裝 Heartbeat 套件(建議使用 yum 工具安裝，免除相依性套件問題)

```
# yum -y localinstall heartbeat-3.0.4-1.el6.x86_64.rpm
heartbeat-devel-3.0.4-1.el6.x86_64.rpm heartbeat-libs-3.0.4-1.el6.x86_64.rpm
```

III. 新增 Heartbeat 服務於系統中

```
[root@Gluster1 ~]# chkconfig --add heartbeat
[root@Gluster1 ~]# chkconfig heartbeat on
```

IV. Heartbeat 有三個設定檔，分別為 ha.cf、haresources 及 authkkeys，將此三個範例設定檔複制到 /etc/ha.d/ 目錄底下或手動新增。

```
[root@Gluster1 ~]# vim /etc/ha.d/ha.cf
[root@Gluster1 ~]# vim /etc/ha.d/haresources
[root@Gluster1 ~]# vim /etc/ha.d/authkeys
```

V. 編輯 Heartbeat 主要配置檔 ha.cf

```
[root@Gluster1 ~]# vim /etc/ha.d/ha.cf
autojoin none

# Debug messages
```

```
debugfile /var/log/ha-debug

# Other messages
logfile /var/log/ha-log

# Syslog 系統日誌
logfacility local0

#  多久時間確認對方是否存活(單位:秒)
keepalive 2

#  多久時間確認對方已完全失去聯繫(單位:秒)
deadtime 10

#  連續多久時間聯繫不上後開始警告提示(單位:秒)
warntime 4

#  主機若重開機，等待網路開啟或其他應用程式執行的時間。(單位:秒)
initdead 60


#  使用 694 Port 來作 Heartbeat 監控

udpport 694


#  採用網路卡 eth1 的 UDP 廣播來發送 heartbeat 訊息

#bcast eth1


#  採用網路卡 eth1 的 UDP 單播  來發送 heartbeat 訊息，IP 為對方 IP 位置。建議採用單

播，避免多組 cluster 主機都會看到對方節點(第二台主機則填對方 IP)。

ucast eth0 172.20.144.84
ucast eth1 1.1.1.2


#  若 Primay 異常修復完畢，是否需要從 Secondary 自動切換回 Primary。建議設定 off。

auto_failback off
```

```
# 節點 1 與節點 2，必須要與 uname -n 指令得到的名稱一致。

node Gluster1
node Gluster2

ping 172.20.144.254

respawn hacluster /usr/lib64/heartbeat/ipfail
respawn hacluster /usr/lib64/heartbeat/dopd
apiauth dopd gid=haclient uid=hacluster
```

VI. 編輯 Heartbeat 認證資訊檔 authkeys (兩台主機 authkeys 內容需一致)

i. **Authkeys 為 Cluster 節點間相互認證的密碼，也就是說主機必須擁有此密碼才可加入該 Cluster 群組。Authkeys 共有三種認証方式，分別為 crc、md5 及 sha1，依安全性等級區分，sha1 最不易破解，其次 md5，最後為 crc。**

ii. **Authkeys 的權限必須為 600**

iii. **兩台主機 authkeys 內容需一致**

```
[root@DRBD-1 ~]# (echo -ne "auth 1\n1 sha1 "; echo $RANDOM | openssl sha1 | awk '{print $2}') > /etc/ha.d/authkeys
[root@Gluster1 ~]# chmod 600 /etc/ha.d/authkeys
[root@Gluster1 ~]# scp /etc/ha.d/authkeys root@Gluster2:/etc/ha.d/
```

```
# authkeys 開啟內容如下

auth 1
1 sha1 ccde56f52bc06bcda0918cfa9439f5c91d941de6
```

VII. 編輯 Heartbeat 資源檔 haresources (兩台主機內容需一致)

i. **Haresources 每一行代表一個資源組**

ii. 資源組的啟動順序是由左至右，關閉順序是由右往左。

iii. Script 的參數是由 :: 來傳遞及分隔。

iv. 每一個資源則以空格間隔

```
[root@Gluster1 ~]# vim /etc/ha.d/haresources
Gluster1 172.20.144.80
```

參數說明:	
Gluster1	指定 Primary 節點
172.20.144.80	Cluster IP 位址

VIII. 啟動 Heartbeat 服務 (兩台主機)

```
[root@Gluster1 ~]# service heartbeat start; ssh Gluster2 service heartbeat start
[root@Gluster1 ~]# service heartbeat status ; ssh Gluster2 service heartbeat status
heartbeat OK [pid 3352 et al] is running on Gluster1 [Gluster1]...
heartbeat OK [pid 3003 et al] is running on Gluster2 [Gluster2]...
```

IX. 啟動完 Heartbeat 服務後，確認 Primary 主機是否具備以下狀態。

i. Heartbeart 新增一組 Cluster IP

```
[root@Gluster1 ~]# ifconfig
eth0:0      Link encap:Ethernet  HWaddr 00:50:56:BD:45:CC
            inet addr:172.20.144.200  Bcast:172.20.144.255  Mask:255.255.255.0
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```