

ucloud server

GlusterFS

설정 가이드

v1.1

2013. 6

목차

1. 개 요.....	4
1.1. GlusterFS 개요.....	4
1.2. ucloud GlusterFS 구성.....	5
2. GlusterFS 서버 구성.....	6
2.1. GlusterFS 서버 VM 생성.....	6
2.1.1. VM 생성.....	6
ucloud Marketplace 의 상품 신청 기능을 이용하여 신규 GlusterFS Server VM 을 생성.....	6
한다.....	6
2.2. 디스크 추가.....	7
2.2.1. 디스크 생성.....	7
2.2.2. 디스크 파티션 설정.....	7
2.2.3. 포맷 및 마운트.....	9
2.2.4. fstab 수정.....	9
2.3. GlusterFS 서버 설정.....	10
2.3.1. Trusted Storage Pool 구성.....	10
2.3.2. Storage Pool 에서 서버 제거.....	10
2.3.3. Volume 생성.....	11
2.3.4. Volume 확인 및 시작.....	14
2.3.5. Volume 제거.....	14
2.4. GlusterFS 관리.....	15
2.4.1. Volume 옵션.....	15
2.4.2. Volume 확장.....	15
2.4.3. Volume 에 속한 Brick 제거.....	15

2.4.4.	Brick 교체(Migration)	16
2.4.5.	Volume Rebalancing	16
3.	GlusterFS 클라이언트 설정	17
3.1.	GlusterFS Native 클라이언트 설치 및 마운트	17
3.1.1.	CentOS 5.4 - 64bit	17
3.1.2.	CentOS 5.4 - 32bit	18
3.1.3.	Debian 6.0 - 64bit	19
3.1.4.	Debian 6.0 - 32bit	20
3.1.5.	Ubuntu (10.04 / 11.04) - 64bit	22
3.1.6.	Ubuntu (10.04 / 10.10 / 11.04) - 32bit	23
3.1.7.	Fedora 13 - 64bit	25
3.2.	NFS Mount	26

1. 개요

본 문서는 ucloud server 상에서 VM 기반의 GlusterFS 를 구성하려는 사용자에게 제공되는 가이드 문서이다.

1.1. GlusterFS 개요

GlusterFS 란 수천 PetaByte 급의 대용량에, 수천 개의 클라이언트가 접속하여 사용 가능한 scale-out 방식의 분산 파일 시스템이다.

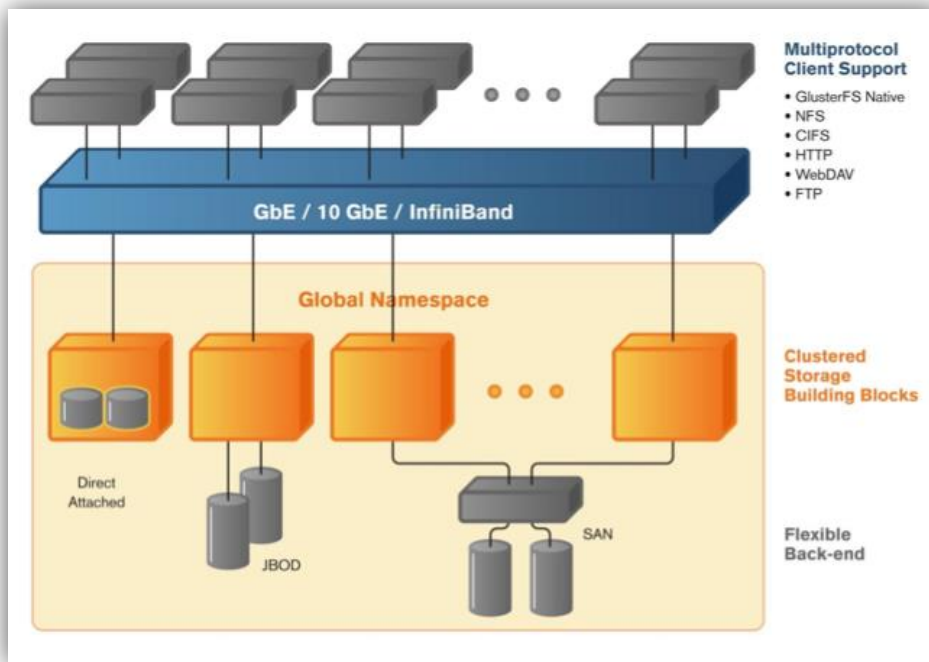


그림 1. GlusterFS 시스템 구성도

GlusterFS 는 기존의 분산 파일 시스템에 비해 비교적 구성이 간단하며, 대용량 / 대규모의 I/O 처리 능력이 뛰어나다.

또한 이중화(Replication) 구성으로 인해 안전성이 뛰어나고, 여러 OS 와의 호환성 등이 우수하기 때문에 다양한 클라우드 인프라 구성에 많이 이용되고 있다.

1.2. ucloud GlusterFS 구성

ucloud Marketplace 의 상품 신청 기능을 이용하여 GlusterFS Server 를 쉽고 빠르게 구성할 수 있다.

URL: https://ucloudbiz.olleh.com/portal/ktcloudportal.epc.productintro.ucloud_server_image.html



그림 2. ucloud Marketplace 화면

생성된 GlusterFS Server VM 은 같은 계정 내 VM 에서 접속하여 사용할 수 있다.

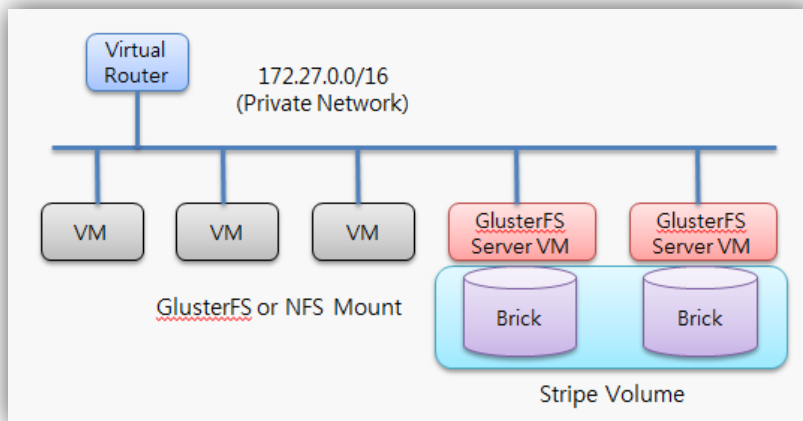


그림 3. ucloud GlusterFS 시스템 구성도

2. GlusterFS 서버 구성

2.1. GlusterFS 서버 VM 생성

2.1.1. VM 생성

ucloud Marketplace의 상품 신청 기능을 이용하여 신규 GlusterFS Server VM을 생성한다.

분류	이미지명	종류	가격	
			월요금제	시간요금제
머신이미지	Centos + GlusterFS	Centos 5.4 64bit	무료	무료

cloud MAS 사용여부 예 아니오

그림 4. ucloud Marketplace 에서 GlusterFS VM 생성 화면

2.2. 디스크 추가

2.2.1. 디스크 생성

ucloud server 관리 콘솔에서 GlusterFS Server VM 에 추가 디스크를 생성한다.



적용 서버	서버 : server01 (4784f32b-2644-47e9-8cd5-76799cc910e5)
스토리지 가격	10GB당 700원
스토리지 용량	10GB
스토리지 명	<input type="text"/> 중복체크 * 영문만 입력 가능합니다. (총 25Bbytes)
월 결제 금액	700원/월 (부가세 별도)

그림 5. ucloud server 관리 콘솔에서 추가 디스크 신청 화면

2.2.2. 디스크 파티션 설정

추가한 디스크를 확인한다.

```
# fdisk -l

Disk /dev/xvda: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1    *           1           13        104391   83  Linux
/dev/xvda2             14          2610       20860402+  8e  Linux LVM

Disk /dev/xvdb: 85.8 GB, 85899345920 bytes // 추가 디스크 확인
255 heads, 63 sectors/track, 10443 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/xvdb doesn't contain a valid partition table
```

디스크 파티션을 생성한다.

```
# fdisk /dev/xvdb

Command (m for help): n // 새 파티션 생성
Command action
    e extended
    p primary partition (1-4)
p // Primary 파티션으로 선택

Partition number (1-4): 1 // 파티션 번호(1) 선택
First cylinder (1-10443, default 1): // 디스크 전체를 생성하는 파티션에 할당
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-10443, default 10443):
Using default value 10443
```

생성한 파티션을 확인한다.

```
Command (m for help): p // 생성한 파티션 확인

Disk /dev/xvdb: 85.8 GB, 85899345920 bytes
255 heads, 63 sectors/track, 10443 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

    Device Boot      Start         End      Blocks   Id  System
/dev/xvdb1          1         10443    83883366   83  Linux
```

설정된 파티션 정보를 저장하고 종료한다.

```
Command (m for help): w // 파티션 정보 저장
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```


2.2.3. 포맷 및 마운트

파티션이 생성된 디스크를 EXT3 파일 시스템으로 포맷한다.

```
# mkfs.ext3 /dev/xvdb1 // 디스크 포맷
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
10485760 inodes, 20970496 blocks
1048524 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
640 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000

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

This filesystem will be automatically checked every 31 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

포맷한 볼륨을 GlusterFS Server VM 에 마운트 한다.

```
# mkdir /data // 추가 디스크를 마운트할 /data 디렉토리 생성
# mount /dev/xvdb1 /mnt // 볼륨 마운트
```

2.2.4. fstab 수정

NFS Server VM 이 재부팅 되더라도 추가한 디스크가 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 내용을 추가한다.

```
/dev/xvdb1 /data ext3 defaults 1 1
```

2.3. GlusterFS 서버 설정

2.3.1. Trusted Storage Pool 구성

GlusterFS 를 사용하기 위해서는 가장 먼저 Storage Pool 을 구성해야 한다. Storage Pool 이란 GlusterFS 서비스를 제공하기 위한 서버들의 집합이며 이 때 각 서버들을 peer 라고 부른다.

다음은 4 개의 GlusterFS 서버를 생성 후 하나의 Storage Pool 로 구성하는 예제이다. 먼저 서버로 사용할 VM 들을 생성 후 (2.1 / 2.2 참조) 다음 명령을 통해 Storage Pool 에 등록한다.

```
# gluster peer probe 172.27.0.6 // IP 가 아닌 Hostname(Domain)을 입력해도 상관 없다.  
# gluster peer probe 172.27.0.7  
# gluster peer probe 172.27.0.8
```

위 명령은 어떠한 서버에서 실행하더라도 상관 없지만, 해당 명령을 수행하는 서버에서는 자신의 서버를 따로 등록할 필요 없다.

다음 명령을 통해 Storage Pool 에 등록된 서버 목록을 확인할 수 있다.

```
# gluster peer status // 등록된 peer 정보 확인  
  
Number of Peers: 3  
  
Hostname: 172.27.0.6  
Uuid: 4e911d81-6ccd-495a-b64c-4a383ceb42ff  
State: Peer in Cluster (Connected)  
  
Hostname: 172.27.0.7  
Uuid: 3d65ce0f-55ed-409d-8bc0-b1de0b857443  
State: Peer in Cluster (Connected)  
  
Hostname: 172.27.0.8  
Uuid: a7bbdb6a-114b-4807-8226-70b686bbf104  
State: Peer in Cluster (Connected)
```

2.3.2. Storage Pool에서 서버 제거

만약 Storage Pool 에 등록된 서버를 제거하고 싶을 경우 다음 명령을 수행한다.

```
# gluster peer detach 172.27.0.8
```

2.3.3. Volume 생성

GlusterFS 에서는 볼륨(Clustered Storage Building Blocks)의 구성원으로 Brick 이라는 개념이 있다.

Brick 이란 GlusterFS 서버 안의 특정한 폴더를 말하며, 이런 여러 개의 Brick 들이 하나의 클러스터로 묶여 볼륨으로 제공된다. (<서버 IP>:/<폴더 이름>)

이 때 볼륨 안에 속한 Brick 들을 어떠한 방식으로 구성했는지에 따라 다양한 타입의 볼륨을 생성할 수 있다.

Distributed Volume

GlusterFS 에서 기본적으로 제공하고 있는 볼륨 타입으로 모든 파일들은 각각의 Brick 에 분산되어 저장된다.

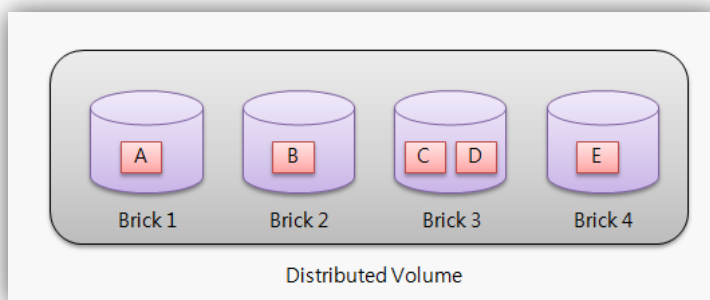


그림 6. Distributed Volume 구성 예

다음 명령을 통해 Distributed Volume 을 생성할 수 있다.

```
# gluster volume create dist_vol 172.27.0.5:/data 172.27.0.6:/data 172.27.0.7:/data  
172.27.0.8:/data
```

Replicated Volume

만약 Brick 여러 개를 이중화(Replication)하여 좀 더 안전성을 향상시키고 싶다면 Replicated 방식으로 구성하면 된다.

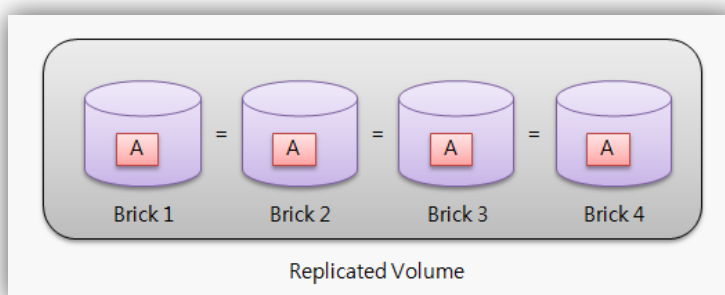


그림 6. Replicated Volume 구성 예

이 방식을 이용하게 되면 Disk I/O 성능이 조금 저하되지만 특정 서버(Brick)에 장애가 발생하더라도 데이터를 안전하게 보존할 수 있다.
만약 각 Brick 간 데이터 불일치 현상이 발생하더라도 자체 Self-Healing 기능을 통해 동기화가 이루어진다.

다음 명령을 통해 Replicated Volume 을 생성할 수 있다.

```
# gluster volume create repl_vol replica 4 transport tcp 172.27.0.5:/data 172.27.0.6:/data  
172.27.0.7:/data 172.27.0.8:/data
```

Striped Volume

Stripe 방식으로 볼륨을 구성하게 되면 파일들은 모든 Brick 에 분산되어 저장된다.

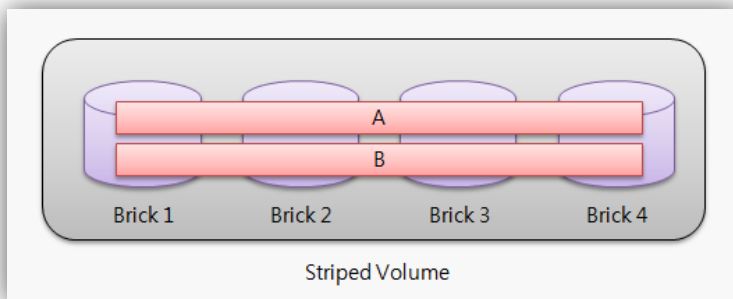


그림 7. Striped Volume 구성 예

Stripe 방식으로 볼륨을 구성할 경우 파일이 모든 Brick 에 분산되어 저장되기 때문에 큰 파일에 대한 I/O 성능을 향상시킬 수 있다.

하지만 Brick 이 하나라도 죽게 되면 모든 파일이 깨지게 되기 때문에 **안전성이 떨어진다.**

다음 명령을 통해 Striped Volume 을 생성할 수 있다.

```
# gluster volume create stri_vol stripe 4 transport tcp 172.27.0.5:/data 172.27.0.6:/data  
172.27.0.7:/data 172.27.0.8:/data
```

Replicated + Distributed Volume

Replicated 방식과 Distributed 방식을 혼합하여 볼륨을 구성할 수 있다.

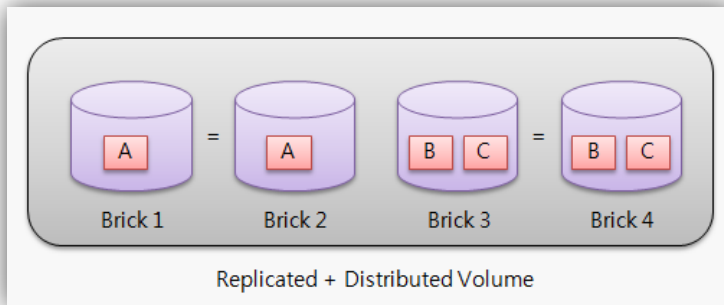


그림 8. Replicated + Distributed Volume 구성 예

위 방식으로 볼륨을 구성하게 되면 Brick 2 개는 서로 이중화되고, 그 이중화된 Brick 들이 또 다시 Distributed 방식으로 묶인다. (RAID 1+0 방식과 동일)

다음 명령을 통해 Replicated + Distributed Volume 을 생성할 수 있다.

```
# gluster volume create repl_dist_vol replica 2 transport tcp 172.27.0.5:/data 172.27.0.6:/data  
172.27.0.7:/data 172.27.0.8:/data
```

Striped + Distributed Volume

Striped 방식 역시 Distributed 방식과 혼합하여 볼륨으로 구성할 수 있다.

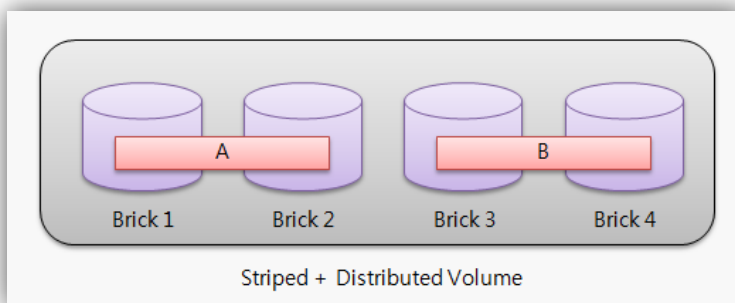


그림 9. Striped + Distributed Volume 구성 예

다음 명령을 통해 Striped + Distributed Volume 을 생성할 수 있다.

```
# gluster volume create stri_dist_vol stripe 2 transport tcp 172.27.0.5:/data 172.27.0.6:/data  
172.27.0.7:/data 172.27.0.8:/data
```

2.3.4. Volume 확인 및 시작

생성된 볼륨 정보는 다음 명령을 통해 확인할 수 있다.

```
# gluster volume info repl_dist_vol // all 을 입력하면 모든 볼륨 정보를 확인할 수 있다.  
  
Volume Name: repl_dist_vol  
Type: Distributed-Replicate  
Status: Created  
Number of Bricks: 2 x 2 = 4  
Transport-type: tcp  
Bricks:  
Brick1: 172.27.0.5:/data  
Brick2: 172.27.0.6:/data  
Brick3: 172.27.0.7:/data  
Brick4: 172.27.0.8:/data
```

볼륨 생성 후 클라이언트에서 접속하여 사용하기 위해서는 볼륨을 시작 상태로 변경해야 한다.

```
# gluster volume start repl_dist_vol
```

2.3.5. Volume 제거

만약 사용 중인 볼륨을 제거하고 싶다면 먼저 볼륨을 중지 시킨다.

```
# gluster volume stop repl_dist_vol
```

이 후 다음 명령을 통해 볼륨을 제거하면 된다.

```
# gluster volume delete repl_dist_vol
```

2.4. GlusterFS 관리

2.4.1. Volume 옵션

GlusterFS 는 다양한 볼륨 옵션 정보를 가지고 있다. (아래 URL 참조)

http://download.gluster.com/pub/gluster/glusterfs/3.2/Documentation/AG/html/chap-Administration_Guide-Managing_Volumes.html#sect-Administration_Guide-Managing_Volumes-Tuning

위 볼륨 옵션 수정을 통해 성능을 튜닝하거나 보안 설정 등을 할 수 있다.

다음은 특정 볼륨에 대한 클라이언트 접속 권한 설정을 하는 예제이다.

```
# gluster volume set repl_dist_vol auth.allow 172.27.*
```

2.4.2. Volume 확장

만약 볼륨 크기가 부족해 확장이 필요할 경우 새롭게 GlusterFS 서버 VM 을 생성 후 Storage Pool 에 등록한다. (2.3.1 참조)

이 후 다음 명령을 통해 볼륨에 Brick 을 추가하면 된다.

```
# gluster volume add-brick dist_vol 172.27.0.9:/data
```

만약 Replicated + Distributed 볼륨이나 Striped + Distributed 볼륨일 경우 Replication 또는 Stripe 로 묶은 단위 만큼 Brick 을 추가해야 한다.

예로 2 Replicated + 2 Distributed 볼륨을 경우 2 Brick 씩 쌍으로 추가해야 한다.

```
# gluster volume add-brick repl_dist_vol 172.27.0.9:/data 172.27.0.10:/data
```

2.4.3. Volume에 속한 Brick 제거

만약 볼륨에 속한 Brick 을 제거하고 싶을 경우 다음 명령을 통해 수행할 수 있다.

```
# gluster volume remove-brick dist_vol 172.27.0.9:/data
```

하지만 Brick 을 제거할 경우 **데이터 손실이 발생할** 수 있기 때문에 권장하지 않으며, Stripe 볼륨은 Brick 을 제거할 수 없다.

또한 Replicated + Distributed 볼륨이나 Striped + Distributed 볼륨 같은 경우 같이 묶인 단위로 Brick 을 제거해야 한다.

2.4.4. Brick 교체(Migration)

만약 기존에 사용하던 Brick 을 새로운 Brick 으로 교체하고 싶을 경우 마이그레이션 기능을 이용하면 된다.

먼저 새로운 GlusterFS 서버를 생성 후 Storage Pool 에 등록한다.

이후 다음 명령을 통해 기존 Brick 안에 저장된 데이터를 새로운 Brick 으로 이전한다.

```
# gluster volume replace-brick dist_vol 172.27.0.8:/data 172.27.0.9:/data start
```

만약 데이터 마이그레이션을 일시 정지하고 싶다면 다음 명령을 수행한다.

```
# gluster volume replace-brick dist_vol 172.27.0.8:/data 172.27.0.9:/data stop
```

마이그레이션을 취소하고 싶다면 다음 명령을 수행한다.

```
# gluster volume replace-brick dist_vol 172.27.0.8:/data 172.27.0.9:/data abort
```

현재 마이그레이션 상태를 알고 싶다면 다음 명령을 수행한다.

```
# gluster volume replace-brick dist_vol 172.27.0.8:/data 172.27.0.9:/data status
```

마지막으로 모든 데이터 마이그레이션이 완료되면 다음 명령을 통해 Brick 을 완전히 교체한다.

```
# gluster volume replace-brick dist_vol 172.27.0.8:/data 172.27.0.9:/data commit
```

2.4.5. Volume Rebalancing

새로운 Brick 을 추가할 경우 모든 Brick 에 데이터가 고르게 분산되지 않아 성능 저하 문제가 발생할 수 있다.

이 때 Rebalancing 기능을 이용하게 되면 모든 Brick 에 데이터를 고르게 분산 시킬 수 있다.

```
# gluster volume rebalance dist_vol start
```

현재 진행 중인 Rebalancing 정보는 다음 명령을 통해 확인할 수 있다.

```
# gluster volume rebalance dist_vol status
```

만약 Rebalancing 작업을 중지하고 싶다면 다음 명령을 수행한다.

```
# gluster volume rebalance dist_vol stop
```


3. GlusterFS 클라이언트 설정

3.1. GlusterFS Native 클라이언트 설치 및 마운트

3.1.1. CentOS 5.4 - 64bit

CentOS 5.4 - 64bit 같은 경우 클라이언트 설치 파일을 rpm 으로 제공하고 있다.
GlusterFS 서버에서 scp 나 wget 으로 클라이언트 설치 파일을 다운로드 받는다.

- wget 이용

```
# wget http://<server_ip>/centos/glusterfs-core-3.2.6-1.x86_64.rpm  
# wget http://<server_ip>/centos/glusterfs-fuse-3.2.6-1.x86_64.rpm
```

- scp 이용

```
# scp root@<server_ip>:/root/gluster_client/centos/glusterfs-core-3.2.6-1.x86_64.rpm ./  
# scp root@<server_ip>:/root/gluster_client/centos/glusterfs-fuse-3.2.6-1.x86_64.rpm ./
```

다운로드 받은 rpm 파일을 설치한다.

```
# rpm -Uvh glusterfs-core-3.2.6-1.x86_64.rpm  
# rpm -Uvh glusterfs-fuse-3.2.6-1.x86_64.rpm
```

fuse 모듈을 올린다.

```
# modprobe fuse
```

GlusterFS 볼륨을 마운트 한다.

```
# mount -t glusterfs 172.27.0.27:/dist_vol /mnt
```

이 때 서버 IP 는 Storage Pool 에 속한 어떠한 서버라도 상관 없으며, Path 는 GlusterFS 볼륨 이름을 입력한다.

클라이언트가 재부팅이 되더라도 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 정보를 추가한다.

```
172.27.0.5:/dist_vol /mnt glusterfs defaults,_netdev 0 0
```

또한 fuse 모듈도 자동으로 올라오도록 하기 위해 **/etc/rc.modules** 파일에 다음 정보를 추가한다.

```
#!/bin/sh  
modprobe fuse
```

그리고 **/etc/rc.modules** 파일에 대한 실행 권한도 부여한다.

```
# chmod 755 /etc/rc.modules
```

3.1.2. CentOS 5.4 - 32bit

CentOS 5.4 - 32bit 는 소스로 클라이언트를 설치해야 한다.

GlusterFS 서버에서 scp 나 wget 으로 클라이언트 설치 파일을 다운로드 받는다.

- wget 이용

```
# wget http://<server_ip>/src/glusterfs-3.2.6.tar.gz
```

- scp 이용

```
# scp root@<server_ip>:/root/gluster_client/src/glusterfs-3.2.6.tar.gz ./
```

다운로드 받은 소스 파일을 압축 해제 한다.

```
# tar xvfz glusterfs-3.2.6.tar.gz  
# cd glusterfs-3.2.6
```

소스 파일을 설치하기 위한 패키지를 설치한다.

```
# yum install gcc flex bison fuse python python-ctypes
```

소스 파일을 설치한다.

```
# ./configure  
# make  
# make install
```

fuse 모듈을 올린다.

```
# modprobe fuse
```

GlusterFS 볼륨을 마운트 한다.

```
# mount -t glusterfs 172.27.0.27:/dist_vol /mnt
```

이 때 서버 IP 는 Storage Pool 에 속한 어떠한 서버라도 상관 없으며, Path 는 GlusterFS 볼륨 이름을 입력한다.

클라이언트가 재부팅이 되더라도 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 정보를 추가한다.

```
172.27.0.5:/dist_vol /mnt glusterfs defaults,_netdev 0 0
```

또한 fuse 모듈도 자동으로 올라오도록 하기 위해 **/etc/rc.modules** 파일에 다음 정보를 추가한다.

```
#!/bin/sh
modprobe fuse
```

그리고 **/etc/rc.modules** 파일에 대한 실행 권한도 부여한다.

```
# chmod 755 /etc/rc.modules
```

3.1.3. Debian 6.0 - 64bit

Debian 6.0 - 64bit 같은 경우 클라이언트 설치 파일을 deb 파일로 제공하고 있다. GlusterFS 서버에서 scp 나 wget 으로 클라이언트 설치 파일을 다운로드 받는다.

- wget 이용

```
# wget http://<server_ip>/debian/glusterfs_3.2.6-1_amd64.deb
```

- scp 이용

```
# scp root@<server_ip>:/root/gluster_client/debian/glusterfs_3.2.6-1_amd64.deb ./
```

주의: Debian 6.0 같은 경우 mirror 사이트가 잘못되어 패키지 설치가 실패할 수 있으니 **/etc/apt/source.list** 를 열어 mirror 사이트 정보를 다음으로 대체한다.

```
deb http://ftp.daum.net/debian/ squeeze main contrib
deb-src http://ftp.daum.net/debian/ squeeze main contrib
deb http://security.debian.org/ squeeze/updates main contrib
deb-src http://security.debian.org/ squeeze/updates main contrib
deb http://ftp.daum.net/debian/ squeeze-updates main contrib
deb-src http://ftp.daum.net/debian/ squeeze-updates main contrib
```

이후 다음 명령을 수행하여 변경된 mirror 사이트 정보를 적용해야 한다.

```
# apt-get update
```

클라이언트 설치에 필요한 패키지를 설치한다.

```
# apt-get install libibverbs1 libibverbs-dev
```

다운로드 받은 deb 파일을 설치한다.

```
# dpkg -i glusterfs_3.2.6-1_amd64.deb
```

fuse 모듈을 올린다.

```
# modprobe fuse
```

GlusterFS 볼륨을 마운트 한다.

```
# mount -t glusterfs 172.27.0.27:/dist_vol /mnt
```

이 때 서버 IP 는 Storage Pool 에 속한 어떠한 서버라도 상관 없으며, Path 는 GlusterFS 볼륨 이름을 입력한다.

클라이언트가 재부팅이 되더라도 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 정보를 추가한다.

```
172.27.0.5:/dist_vol /mnt glusterfs defaults,_netdev 0 0
```

또한 fuse 모듈도 자동으로 올라오도록 하기 위해 **/etc/modules** 파일에 다음 정보를 추가한다.

```
fuse
```

3.1.4. Debian 6.0 - 32bit

CentOS 5.4 - 32bit 는 소스로 클라이언트를 설치해야 한다.

GlusterFS 서버에서 scp 나 wget 으로 클라이언트 설치 파일을 다운로드 받는다.

- wget 이용

```
# wget http://<server_ip>/src/glusterfs-3.2.6.tar.gz
```

- scp 이용

```
# scp root@<server_ip>:/root/gluster_client/src/glusterfs-3.2.6.tar.gz ./
```

다운로드 받은 소스 파일을 압축 해제 한다.

```
# tar xvfz glusterfs-3.2.6.tar.gz
```

```
# cd glusterfs-3.2.6
```

주의: Debian 6.0 같은 경우 mirror 사이트가 잘못되어 패키지 설치가 실패할 수 있으니 **/etc/apt/source.list** 를 열어 mirror 사이트 정보를 다음으로 대체한다.

```
deb http://ftp.daum.net/debian/ squeeze main contrib
deb-src http://ftp.daum.net/debian/ squeeze main contrib
deb http://security.debian.org/ squeeze/updates main contrib
deb-src http://security.debian.org/ squeeze/updates main contrib
deb http://ftp.daum.net/debian/ squeeze-updates main contrib
```

```
deb-src http://ftp.daum.net/debian/ squeeze-updates main contrib
```

이후 다음 명령을 수행하여 변경된 mirror 사이트 정보를 적용해야 한다.

```
# apt-get update
```

소스 파일을 설치하기 위한 패키지를 설치한다.

```
# apt-get install gcc flex bison make
```

소스 파일을 설치한다.

```
# ./configure  
# make  
# make install
```

GlusterFS 관련 라이브러리를 추가한다.

```
# ln -s /usr/local/lib/libglusterfs.so.0 /usr/lib/libglusterfs.so.0  
# ln -s /usr/local/lib/libgfxdr.so.0 /usr/lib/libgfxdr.so.0  
# ln -s /usr/local/lib/libgfrpc.so.0 /usr/lib/libgfrpc.so.0
```

fuse 모듈을 올린다.

```
# modprobe fuse
```

GlusterFS 볼륨을 마운트 한다.

```
# mount -t glusterfs 172.27.0.27:/dist_vol /mnt
```

이 때 서버 IP 는 Storage Pool 에 속한 어떠한 서버라도 상관 없으며, Path 는 GlusterFS 볼륨 이름을 입력한다.

클라이언트가 재부팅이 되더라도 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 정보를 추가한다.

```
172.27.0.5:/dist_vol /mnt glusterfs defaults,_netdev 0 0
```

또한 fuse 모듈도 자동으로 올라오도록 하기 위해 **/etc/modules** 파일에 다음 정보를 추가한다.

```
fuse
```

3.1.5. Ubuntu (10.04 / 11.04) - 64bit

Ubuntu - 64bit 같은 경우 클라이언트 설치 파일을 deb 파일로 제공하고 있다.
GlusterFS 서버에서 scp 나 wget 으로 클라이언트 설치 파일을 다운로드 받는다.

- wget 이용

```
# wget http://<server_ip>/ubuntu/10.04/glusterfs_3.2.6-1_amd64.deb
# wget http://<server_ip>/ubuntu/11.04/glusterfs_3.2.6-1_amd64.deb
```

- scp 이용

```
# scp root@<server_ip>:/root/gluster_client/ubuntu/10.04/glusterfs_3.2.6-1_amd64.deb ./
# scp root@<server_ip>:/root/gluster_client/ubuntu/11.04/glusterfs_3.2.6-1_amd64.deb ./
```

위 설치 파일 중 버전에 맞게 다운로드 받으면 된다.

주의: Ubuntu 같은 경우 mirror 사이트가 잘못되어 패키지 설치가 실패할 수 있으니 **/etc/apt/source.list** 를 열어 mirror 사이트 정보를 다음으로 대체한다.

```
deb http://ftp.daum.net/ubuntu lucid main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-backports main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid-backports main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-proposed main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid-proposed main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-security main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid-security main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-updates main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid-updates main multiverse restricted universe
```

만약 11.04 버전을 사용하는 경우 lucid 단어를 natty 로 교체한다.

이후 다음 명령을 수행하여 변경된 mirror 사이트 정보를 적용해야 한다.

```
# apt-get update
```

필요 패키지 설치 및 클라이언트에 설치된 패키지 버전을 업그레이드 한다.

```
# apt-get install nfs-common
# apt-get -f install
```

다운로드 받은 deb 파일을 설치한다.

```
# dpkg -i glusterfs_3.2.6-1_amd64.deb
```

fuse 모듈을 올린다.

```
# modprobe fuse
```

GlusterFS 볼륨을 마운트 한다.

```
# mount -t glusterfs 172.27.0.27:/dist_vol /mnt
```

이 때 서버 IP 는 Storage Pool 에 속한 어떠한 서버라도 상관 없으며, Path 는 GlusterFS 볼륨 이름을 입력한다.

클라이언트가 재부팅이 되더라도 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 정보를 추가한다.

```
172.27.0.5:/dist_vol /mnt glusterfs defaults,_netdev 0 0
```

또한 fuse 모듈도 자동으로 올라오도록 하기 위해 **/etc/modules** 파일에 다음 정보를 추가한다.

```
fuse
```

3.1.6. Ubuntu (10.04 / 10.10 / 11.04) - 32bit

Ubuntu - 32bit 는 소스로 클라이언트를 설치해야 한다.

GlusterFS 서버에서 scp 나 wget 으로 클라이언트 설치 파일을 다운로드 받는다.

- wget 이용

```
# wget http://<server_ip>/src/glusterfs-3.2.6.tar.gz
```

- scp 이용

```
# scp root@<server_ip>:/root/gluster_client/src/glusterfs-3.2.6.tar.gz ./
```

다운로드 받은 소스 파일을 압축 해제 한다.

```
# tar xvfz glusterfs-3.2.6.tar.gz
```

```
# cd glusterfs-3.2.6
```

주의: Ubuntu 같은 경우 mirror 사이트가 잘못되어 패키지 설치가 실패할 수 있으니 **/etc/apt/source.list** 를 열어 mirror 사이트 정보를 다음으로 대체한다.

```
deb http://ftp.daum.net/ubuntu lucid main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-backports main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid-backports main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-proposed main multiverse restricted universe
```

```
deb-src http://ftp.daum.net/ubuntu lucid-proposed main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-security main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid-security main multiverse restricted universe
deb http://ftp.daum.net/ubuntu lucid-updates main multiverse restricted universe
deb-src http://ftp.daum.net/ubuntu lucid-updates main multiverse restricted universe
```

만약 11.04 버전을 사용하는 경우 lucid 단어를 natty 로 교체한다.

소스 파일을 설치하기 위한 패키지를 설치한다.

```
# apt-get install gcc flex bison make
```

소스 파일을 설치한다.

```
# ./configure
# make
# make install
```

GlusterFS 관련 라이브러리를 추가한다.

```
# ln -s /usr/local/lib/libglusterfs.so.0 /usr/lib/libglusterfs.so.0
# ln -s /usr/local/lib/libgfsxdr.so.0 /usr/lib/libgfsxdr.so.0
# ln -s /usr/local/lib/libgfrpc.so.0 /usr/lib/libgfrpc.so.0
```

fuse 모듈을 올린다.

```
# modprobe fuse
```

GlusterFS 볼륨을 마운트 한다.

```
# mount -t glusterfs 172.27.0.27:/dist_vol /mnt
```

이 때 서버 IP 는 Storage Pool 에 속한 어떠한 서버라도 상관 없으며, Path 는 GlusterFS 볼륨 이름을 입력한다.

클라이언트가 재부팅이 되더라도 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 정보를 추가한다.

```
172.27.0.5:/dist_vol /mnt glusterfs defaults,_netdev 0 0
```

또한 fuse 모듈도 자동으로 올라오도록 하기 위해 **/etc/modules** 파일에 다음 정보를 추가한다.

```
fuse
```


3.1.7. Fedora 13 - 64bit

Fedora 13 - 64bit 는 소스로 클라이언트를 설치해야 한다.

GlusterFS 서버에서 scp 나 wget 으로 클라이언트 설치 파일을 다운로드 받는다.

- wget 이용

```
# wget http://<server_ip>/src/glusterfs-3.2.6.tar.gz
```

- scp 이용

```
# scp root@<server_ip>:/root/gluster_client/src/glusterfs-3.2.6.tar.gz ./
```

다운로드 받은 소스 파일을 압축 해제 한다.

```
# tar xvfz glusterfs-3.2.6.tar.gz
```

```
# cd glusterfs-3.2.6
```

소스 파일을 설치하기 위한 패키지를 설치한다.

```
# yum install gcc flex bison make
```

소스 파일을 설치한다.

```
# ./configure
```

```
# make
```

```
# make install
```

fuse 모듈을 올린다.

```
# modprobe fuse
```

GlusterFS 볼륨을 마운트 한다.

```
# mount -t glusterfs 172.27.0.27:/dist_vol /mnt
```

이 때 서버 IP 는 Storage Pool 에 속한 어떠한 서버라도 상관 없으며, Path 는 GlusterFS 볼륨 이름을 입력한다.

클라이언트가 재부팅이 되더라도 자동으로 마운트 되도록 **/etc/fstab** 파일에 다음 정보를 추가한다.

```
172.27.0.5:/dist_vol /mnt glusterfs defaults,_netdev 0 0
```

또한 fuse 모듈도 자동으로 올라오도록 하기 위해 **/etc/rc.modules** 파일에 다음 정보를 추가한다.

```
#!/bin/sh
modprobe fuse
```

그리고 **/etc/rc.modules** 파일에 대한 실행 권한도 부여한다.

```
# chmod 755 /etc/rc.modules
```

3.2. NFS Mount

클라이언트에서 GlusterFS 볼륨을 마운트 시 위에서 설명한 GlusterFS Native 클라이언트가 아닌 Linux의 NFS 클라이언트를 그대로 이용해도 된다.

```
# mount -t nfs 172.27.0.27:/dist_vol /mnt
```

하지만 NFS 방식으로 마운트할 경우 모든 데이터 트래픽이 NFS 서버에 몰리기 때문에 Native 클라이언트에 비해 성능이 좋지 않다.