

MPIO (Multipath I/O)

1. multipath.conf 파일 확인

```
blacklist {  
    devnode "*"   
}
```

-> 시스템 기본 장치에 대한 black list 설정 부분 주석이 풀려 있음

```
defaults {  
    user_friendly_names yes   
}
```

-> wwid 대신 friendly name을 사용하여 alias를 지정할수 있도록 설정, 주석이 풀려 있음

```
defaults {  
    udev_dir                /dev  
    polling_interval        10  
    selector                "round-robin 0"  
    path_grouping_policy    multibus  
    getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"  
    prio_callout            /bin/true  
    path_checker            readsector0  
    rr_min_io               100  
    max_fds                 8192  
    rr_weight               priorities  
    failback                immediate  
    no_path_retry           fail  
    user_friendly_names     yes  
}
```

-> default 설정에 대한 예제. 주석처리가 되어 있음. 주석을 풀어준다.

```
blacklist {  
#    wwid 26353900f02796769  
    devnode "(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"  
    devnode "hd[a-z]"  
    devnode "cciss!c[0-9]d[0-9]*"  
}
```

-> blacklist 설정에 대한 예제. 주석처리가 되어 있음. wwid xxx 라인을 제외하고 주석을 풀어준다

-> HP 장비일 경우는 devnode "cciss!c[0-9]d[0-9]*" 부분을 추가해 준다.

```
multipaths {  
#    multipath {
```

```

#           wwid           3600508b4000156d700012000000b0000
#           alias          yellow
#           path_grouping_policy  multibus
#           path_checker    readsector0
#           path_selector   "round-robin 0"
#           failback        manual
#           rr_weight       priorities
#           no_path_retry   5
#       }
multipath {
           wwid           1DEC_____321816758474
           alias          red
}
}

```

-> multipath 설정에 대한 예제. 주석처리 되어 있음. wwid와 alias를 이용해서 multipath를 설정을 추가하게 된다. 주석 해제.

2. SAN Disk 확인

SAN 스토리지가 연결된 경우, 아래와 같이 local disk 이외에 수많은 disk가 보이게 된다. 여기서 SAN 할당 영역 LUN(12.1G)을 제외하고 local disk 부분, SAN 스토리지의 OS 영역 등을 blacklist로 추가하여 multipath에서 제외 시킨다.

```
[root@localhost etc]#fdisk -l
```

```
Disk /dev/sda: 298.9 GB, 298999349248 bytes
255 heads, 63 sectors/track, 36351 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	261	2096451	83	Linux
/dev/sda2		262	1566	10482412+	83	Linux
/dev/sda3		1567	2610	8385930	82	Linux swap / Solaris
/dev/sda4		2611	36351	271024582+	5	Extended
/dev/sda5		2611	6526	31455238+	83	Linux
/dev/sda6		6527	9137	20972826	83	Linux
/dev/sda7		9138	11487	18876343+	83	Linux
/dev/sda8		11488	11748	2096451	83	Linux
/dev/sda9		11749	13243	12008556	83	Linux
/dev/sda10		13244	25693	100004593+	83	Linux

```
Disk /dev/sdb: 298.9 GB, 298999349248 bytes
255 heads, 63 sectors/track, 36351 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	36351	291989376	8e	Linux LVM

Disk /dev/sdc (Sun disk label): 15 heads, 128 sectors, 1 cylinders
 Units = cylinders of 1920 * 512 bytes

Device	Flag	Start	End	Blocks	Id	System
/dev/sdc3	u	0	1	960	5	Whole disk
/dev/sdc7		0	1	960	4	SunOS usr

Disk /dev/sdc7 (Sun disk label): 15 heads, 128 sectors, 1 cylinders
 Units = cylinders of 1920 * 512 bytes

Device	Flag	Start	End	Blocks	Id	System
/dev/sdc7p3	u	0	1	960	5	Whole disk
/dev/sdc7p7		0	1	960	4	SunOS usr

Disk /dev/sdd: 12.1 GB, 12178882560 bytes
 64 heads, 32 sectors/track, 11614 cylinders
 Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdd doesn't contain a valid partition table

(중간생략)

Disk /dev/sdr (Sun disk label): 15 heads, 128 sectors, 1 cylinders
 Units = cylinders of 1920 * 512 bytes

Device	Flag	Start	End	Blocks	Id	System
/dev/sdr3	u	0	1	960	5	Whole disk
/dev/sdr7		0	1	960	4	SunOS usr

Disk /dev/sdr7 (Sun disk label): 15 heads, 128 sectors, 1 cylinders
 Units = cylinders of 1920 * 512 bytes

Device	Flag	Start	End	Blocks	Id	System
/dev/sdr7p3	u	0	1	960	5	Whole disk
/dev/sdr7p7		0	1	960	4	SunOS usr

Disk /dev/sds: 12.1 GB, 12178882560 bytes
 64 heads, 32 sectors/track, 11614 cylinders
 Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sds doesn't contain a valid partition table

(중간생략)

Disk /dev/sdz: 2 MB, 2949120 bytes
1 heads, 6 sectors/track, 960 cylinders
Units = cylinders of 6 * 512 = 3072 bytes

Disk /dev/sdz doesn't contain a valid partition table

Disk /dev/sdaa: 12.1 GB, 12178882560 bytes
64 heads, 32 sectors/track, 11614 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdaa doesn't contain a valid partition table

(중간생략)

Disk /dev/sdae doesn't contain a valid partition table

Disk /dev/sdaf: 2 MB, 2949120 bytes
1 heads, 6 sectors/track, 960 cylinders
Units = cylinders of 6 * 512 = 3072 bytes

Disk /dev/sdaf doesn't contain a valid partition table

Disk /dev/sdag: 12.1 GB, 12178882560 bytes
64 heads, 32 sectors/track, 11614 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdag doesn't contain a valid partition table

(중간생략)

Disk /dev/sdak: 2 MB, 2949120 bytes
1 heads, 6 sectors/track, 960 cylinders
Units = cylinders of 6 * 512 = 3072 bytes

Disk /dev/sdak doesn't contain a valid partition table

Disk /dev/sdal: 12.1 GB, 12178882560 bytes
64 heads, 32 sectors/track, 11614 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdal doesn't contain a valid partition table

Disk /dev/sdam: 12.1 GB, 12178882560 bytes
64 heads, 32 sectors/track, 11614 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

Disk /dev/sdam doesn't contain a valid partition table

Disk /dev/sdan: 2 MB, 2949120 bytes
1 heads, 6 sectors/track, 960 cylinders
Units = cylinders of 6 * 512 = 3072 bytes

Disk /dev/sdan doesn't contain a valid partition table

(중간생략)

Disk /dev/dm-3: 12.1 GB, 12178882560 bytes
255 heads, 63 sectors/track, 1480 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/dm-3 doesn't contain a valid partition table

(중간생략)

Disk /dev/dm-18: 12.1 GB, 12178882560 bytes
255 heads, 63 sectors/track, 1480 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Disk /dev/dm-18 doesn't contain a valid partition table
[root@localhost etc]#

-> "fdisk -l | grep Disk" 명령으로 Disk 부분만 확인

[root@localhost etc]#**fdisk -l | grep Disk**

Disk /dev/sdd doesn't contain a valid partition table

(중간생략)

Disk /dev/sdq doesn't contain a valid partition table

Disk /dev/sda: 298.9 GB, 298999349248 bytes

Disk /dev/sdb: 298.9 GB, 298999349248 bytes

Disk /dev/sdc (Sun disk label): 15 heads, 128 sectors, 1 cylinders

Disk /dev/sdc7 (Sun disk label): 15 heads, 128 sectors, 1 cylinders

Disk /dev/sdd: 12.1 GB, 12178882560 bytes

(중간생략)

Disk /dev/sdq: 12.1 GB, 12178882560 bytes

Disk /dev/sdr (Sun disk label): 15 heads, 128 sectors, 1 cylinders

Disk /dev/sdr7 (Sun disk label): 15 heads, 128 sectors, 1 cylinders

Disk /dev/sds doesn't contain a valid partition table

(중간생략)

```

Disk /dev/dm-7 doesn't contain a valid partition table
Disk /dev/sds: 12.1 GB, 12178882560 bytes
(중간생략)
Disk /dev/sdaq: 2 MB, 2949120 bytes
Disk /dev/sdar: 2 MB, 2949120 bytes
Disk /dev/dm-3: 12.1 GB, 12178882560 bytes
Disk /dev/dm-9 doesn't contain a valid partition table
Disk /dev/dm-4: 12.1 GB, 12178882560 bytes
(중간생략)
Disk /dev/dm-18 doesn't contain a valid partition table
Disk /dev/dm-8: 12.1 GB, 12178882560 bytes
(중간생략)
Disk /dev/dm-18: 12.1 GB, 12178882560 bytes
[root@localhost etc]#

```

-> doesn't contain a valid partition table 이라고 나오는 메시지는 각각의 디바이스가 파티션이 나뉘져 있지 않다는 의미. 무시해도 됨
Disk /dev/dm-3 등과 같이 dm- 로 보이는 device 는 두개의 device가 이중화 되어 보이는 device로 실제 device가 아님

3. multipath 구성

```

-> multipath 구성할 실제 디스크 확인
[root@localhost rock]# fdisk -l | grep Disk | grep -v dm- > sdisk
[root@localhost rock]# cat sdisk

```

-> 실제 디스크에서 12.1G 로 할당된 디스크를 제외하고 나머지 디스크를 blacklist로 등록한다.

```

[root@localhost rock]#cat sdisk | grep -v 12.1
Disk /dev/sda: 298.9 GB, 298999349248 bytes
Disk /dev/sdb: 298.9 GB, 298999349248 bytes
Disk /dev/sdc (Sun disk label): 15 heads, 128 sectors, 1 cylinders
Disk /dev/sdc7 (Sun disk label): 15 heads, 128 sectors, 1 cylinders
Disk /dev/sdr (Sun disk label): 15 heads, 128 sectors, 1 cylinders
Disk /dev/sdr7 (Sun disk label): 15 heads, 128 sectors, 1 cylinders
Disk /dev/sdz: 2 MB, 2949120 bytes
Disk /dev/sdaf: 2 MB, 2949120 bytes
Disk /dev/sdak: 2 MB, 2949120 bytes
Disk /dev/sdan: 2 MB, 2949120 bytes
Disk /dev/sdao: 2 MB, 2949120 bytes
Disk /dev/sdap: 2 MB, 2949120 bytes
Disk /dev/sdaq: 2 MB, 2949120 bytes
Disk /dev/sdar: 2 MB, 2949120 bytes

```

-> blacklist로 등록할때는 device 명을 정규 표현식으로 입력해도 되지만 각 device 의 wwid를 확인해서 wwid를 이용해서 blacklist 추가 하는 것을 권장.

wwid 확인은 아래와 같이 할 수 있다.

```
[root@localhost rock]#scsi_id -g -u -s /block/sda
3600605b002356e7013d6f2803da5967a
[root@localhost rock]#scsi_id -g -u -s /block/sdb
3600605b002356e7013d6f2803da5cdb3
[root@localhost rock]#scsi_id -g -u -s /block/sdc
360060480000290103591533030303230
[root@localhost rock]#
```

-> SAN 영역은 실제 Disk는 하나지만 path 이중화로 인해 디스크가 두개로 보이게 된다. wwid가 같은 disk 가 두개씩 존재.

아래와 같이 sort | uniq 하게 되면 Local Disk 영역을 제외하고 나머지가 반으로 줄어든다.

```
[root@localhost rock]#for a in $(cat sdisk | grep -v 12.1|cut -d: -f1 |awk -F/ '{print $3}'|
awk '{print $1}')
> do scsi_id -g -u -s /block/$a
> done
3600605b002356e7013d6f2803da5967a
3600605b002356e7013d6f2803da5cdb3
360060480000290103591533030303230
360060480000290103591533030303230
360060480000290103591533030324334
360060480000290103591533030324335
360060480000290103591533030324336
360060480000290103591533030324334
360060480000290103591533030324337
360060480000290103591533030324335
360060480000290103591533030324336
360060480000290103591533030324337
```

```
[root@localhost rock]#for a in $(cat sdisk | grep -v 12.1|cut -d: -f1 |awk -F/ '{print $3}'|
awk '{print $1}'); do scsi_id -g -u -s /block/$a; done | sort | uniq
360060480000290103591533030303230
360060480000290103591533030324334
360060480000290103591533030324335
360060480000290103591533030324336
360060480000290103591533030324337
3600605b002356e7013d6f2803da5967a
3600605b002356e7013d6f2803da5cdb3
[root@localhost rock]#
```

-> uniq 한 wwid 값을 가지고 /etc/multipath.conf 파일에 blacklist 추가

```
blacklist {
#         wwid 26353900f02796769
```

```

devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
devnode "^hd[a-z]"
devnode "^cciss!c[0-9]d[0-9]*"
wwid 3600605b002356e7013d6f2803da5967a
wwid 3600605b002356e7013d6f2803da5cdb3
wwid 360060480000290103591533030303230
wwid 360060480000290103591533030324334
wwid 360060480000290103591533030324335
wwid 360060480000290103591533030324336
wwid 360060480000290103591533030324337
}

```

-> 마지막으로 12.1G 에 해당하는 디스크의 wwid를 확인하고 sort, uniq 하면 실제 할당된 LUN 갯수만큼 wwid를 확인할 수 있다.

```

[root@localhost rock]#for a in $(cat sdisk | grep 12.1 | cut -d: -f1 | awk -F/ '{print $3}');
do scsi_id -g -u -s /block/$a; done | sort | uniq
360060480000290103591533030353736
360060480000290103591533030353737
360060480000290103591533030353738
360060480000290103591533030353739
360060480000290103591533030353741
360060480000290103591533030353742
360060480000290103591533030353743
360060480000290103591533030353744
360060480000290103591533030353745
360060480000290103591533030353746
360060480000290103591533030353830
360060480000290103591533030353831
360060480000290103591533030353832
360060480000290103591533030353833
360060480000290103591533030353834
360060480000290103591533030353835
[root@localhost rock]#

```

-> 확인된 wwid를 /etc/multipath.conf에 아래와 같이 하나씩 multipath 설정 한다.
alias는 임의로 지정

```

multipaths {
#       multipath {
#           wwid           3600508b4000156d700012000000b0000
#           alias          yellow
#           path_grouping_policy  multibus
#           path_checker    readsector0
#           path_selector   "round-robin 0"

```



```
#         failback          manual
#         rr_weight        priorities
#         no_path_retry    5
#     }
multipath {
    wwid          360060480000290103591533030353736
    alias        data1
}
multipath {
    wwid          360060480000290103591533030353737
    alias        data2
}
multipath {
    wwid          360060480000290103591533030353738
    alias        data3
}
multipath {
    wwid          360060480000290103591533030353739
    alias        data4
}
multipath {
    wwid          360060480000290103591533030353741
    alias        data5
}
multipath {
    wwid          360060480000290103591533030353742
    alias        data6
}
multipath {
    wwid          360060480000290103591533030353743
    alias        data7
}
multipath {
    wwid          360060480000290103591533030353744
    alias        data8
}
multipath {
    wwid          360060480000290103591533030353745
    alias        data9
}
multipath {
    wwid          360060480000290103591533030353746
    alias        data10
}
multipath {
```

```

        wwid          360060480000290103591533030353830
        alias         data11
    }
    multipath {
        wwid          360060480000290103591533030353831
        alias         data12
    }
    multipath {
        wwid          360060480000290103591533030353832
        alias         data13
    }
    multipath {
        wwid          360060480000290103591533030353833
        alias         data14
    }
    multipath {
        wwid          360060480000290103591533030353834
        alias         data15
    }
    multipath {
        wwid          360060480000290103591533030353835
        alias         data16
    }
}

```

-> 설정을 완료한 후에는 multipath 데몬을 시작한다.

```
[root@localhost rock]#chkconfig multipathd on
```

```
[root@localhost rock]#service multipathd start
```

-> blacklist 설정이 잘 됐는지, multipath -v3 | grep blacklist 명령으로 확인

```
[root@localhost raw]#multipath -v3 | grep blacklist
```

```
dm-0: blacklisted
```

```
dm-100: blacklisted
```

```
dm-101: blacklisted
```

```
dm-102: blacklisted
```

(중간생략)

```
ram7: blacklisted
```

```
ram8: blacklisted
```

```
ram9: blacklisted
```

```
360060480000290103591533030324335: blacklisted
```

```
360060480000290103591533030324336: blacklisted
```

```
360060480000290103591533030324334: blacklisted
```

```
360060480000290103591533030324337: blacklisted
```

```
360060480000290103591533030324335: blacklisted
```

360060480000290103591533030324336: blacklisted
360060480000290103591533030324337: blacklisted
3600605b002356e7013d6f2803da5967a: blacklisted
3600605b002356e7013d6f2803da5cdb3: blacklisted
360060480000290103591533030303230: blacklisted
360060480000290103591533030303230: blacklisted
360060480000290103591533030324334: blacklisted

-> multipath 설정이 잘 됐는지, multipath -v2 -ll 명령으로 확인, [size=11G]로 표시되는 device가 실제 LUN 개수와 동일하니 확인 한다.

```
[root@localhost etc]#multipath -v2 -ll
```

```
data16 (360060480000290103591533030353835) dm-13 EMC,SYMMETRIX  
[size=11G][features=0][hwhandler=0][rw]
```

```
W_ round-robin 0 [prio=0][active]
```

```
W_ 7:0:0:32 sdam 66:96 [active][ready]
```

```
W_ 5:0:0:32 sdw 65:96 [active][ready]
```

```
data15 (360060480000290103591533030353834) dm-12 EMC,SYMMETRIX  
[size=11G][features=0][hwhandler=0][rw]
```

```
W_ round-robin 0 [prio=0][active]
```

```
W_ 7:0:0:31 sdal 66:80 [active][ready]
```

```
W_ 5:0:0:31 sds 65:32 [active][ready]
```

```
data14 (360060480000290103591533030353833) dm-11 EMC,SYMMETRIX  
[size=11G][features=0][hwhandler=0][rw]
```

```
W_ round-robin 0 [prio=0][active]
```

```
W_ 7:0:0:30 sdaj 66:48 [active][ready]
```

```
W_ 5:0:0:30 sdq 65:0 [active][ready]
```

```
data13 (360060480000290103591533030353832) dm-10 EMC,SYMMETRIX  
[size=11G][features=0][hwhandler=0][rw]
```

```
W_ round-robin 0 [prio=0][active]
```

```
W_ 7:0:0:29 sdai 66:32 [active][ready]
```

```
W_ 5:0:0:29 sdp 8:240 [active][ready]
```

```
data12 (360060480000290103591533030353831) dm-9 EMC,SYMMETRIX  
[size=11G][features=0][hwhandler=0][rw]
```

```
W_ round-robin 0 [prio=0][active]
```

```
W_ 7:0:0:28 sdah 66:16 [active][ready]
```

```
W_ 5:0:0:28 sdo 8:224 [active][ready]
```

(중간생략)

```
data3 (360060480000290103591533030353738) dm-16 EMC,SYMMETRIX  
[size=11G][features=0][hwhandler=0][rw]
```

```
W_ round-robin 0 [prio=0][active]
```

```
W_ 5:0:0:19 sdf 8:80 [active][ready]
```

```
W_ 7:0:0:19 sdv 65:80 [active][ready]
```

```
data2 (360060480000290103591533030353737) dm-15 EMC,SYMMETRIX  
[size=11G][features=0][hwhandler=0][rw]
```

```
W_ round-robin 0 [prio=0][active]
```

```

W_ 5:0:0:18 sde 8:64 [active][ready]
W_ 7:0:0:18 sdu 65:64 [active][ready]
data1 (360060480000290103591533030353736) dm-14 EMC,SYMMETRIX
[size=11G][features=0][hwhandler=0][rw]
W_ round-robin 0 [prio=0][active]
W_ 5:0:0:17 sdd 8:48 [active][ready]
W_ 7:0:0:17 sdt 65:48 [active][ready]
[root@localhost etc]#

```

-> alias 설정 이름으로 multipath 설정이 되었으며, /dev/mapper 디렉토리 밑에 alias명으로 device가 생겼다.

```

[root@localhost mapper]#ls -l
total 0
brw-rw---- 1 root disk 253, 114 Aug 16 17:01 VG01-ARCH
brw-rw---- 1 root disk 253, 111 Aug 16 17:01 VG01-BACKUP
brw-rw---- 1 root disk 253, 113 Aug 16 17:01 VG01-LOG
brw-rw---- 1 root disk 253, 110 Aug 16 17:01 VG01-ORACLE
brw-rw---- 1 root disk 253, 115 Aug 16 17:01 VG01-WORK
brw-rw---- 1 root disk 253,  0 Aug 16 17:01 VG01-app
brw-rw---- 1 root disk 253,  2 Aug 16 17:01 VG01-log
brw-rw---- 1 root disk 253,  1 Aug 16 17:01 VG01-svc
crw----- 1 root root  10,  63 Aug 12 04:12 control
brw-rw---- 1 root disk 253, 14 Aug 12 12:07 data1
brw-rw---- 1 root disk 253,  7 Aug 12 12:07 data10
brw-rw---- 1 root disk 253,  8 Aug 12 12:07 data11
brw-rw---- 1 root disk 253,  9 Aug 12 12:07 data12
brw-rw---- 1 root disk 253, 10 Aug 12 12:07 data13
brw-rw---- 1 root disk 253, 11 Aug 12 12:07 data14
brw-rw---- 1 root disk 253, 12 Aug 12 12:07 data15
brw-rw---- 1 root disk 253, 13 Aug 12 12:07 data16
brw-rw---- 1 root disk 253, 15 Aug 12 12:07 data2
brw-rw---- 1 root disk 253, 16 Aug 12 12:07 data3
brw-rw---- 1 root disk 253, 17 Aug 12 12:07 data4
brw-rw---- 1 root disk 253, 18 Aug 12 12:07 data5
brw-rw---- 1 root disk 253,  3 Aug 12 12:07 data6
brw-rw---- 1 root disk 253,  4 Aug 12 12:07 data7
brw-rw---- 1 root disk 253,  5 Aug 12 12:07 data8
brw-rw---- 1 root disk 253,  6 Aug 12 12:07 data9
[root@localhost mapper]#

```

Rawdevice 할당

4. VG구성 및 LV 생성

-> /dev/mapper/dataX 디바이스에 대해서, PV를 생성하고 VG, LV를 생성한다.

```
[root@localhost rock]#pvcreate /dev/mapper/data[1-9] /dev/mapper/data1[0-6]
[root@localhost rock]#vgcreate VG10 /dev/mapper/data[1-9] /dev/mapper1[0-6]
```

-> 아래의 shell을 적절하게 수정해서, 클라이언트 요구사항에 맞게 Lvol을 생성한다.
ex)

```
=====
#!/bin/bash

# rvol1 ~ rvol10 1024M LV생성
# -i 옵션 뒤의 숫자는 VG10의 pv 갯수
i=1
while (( $i < 11 ))
do
lvcreate -n rvol$i -L 1024M -i 15 VG10
let i+=1
done

# rvol11 ~ rvol50 2048M LV생성
# -i 옵션 뒤의 숫자는 VG10의 pv 갯수
while (( $i < 51 ))
do
lvcreate -n rvol$i -L 2048M -i 15 VG10
let i+=1
done
=====
```

-> LV 생성을 완료하면 /dev/mapper 디렉토리에 생성된 LV를 확인 할 수 있다.

```
[root@localhost mapper]#ls
VG01-ARCH VG10-rvol9 VG12-rvol4 VG13-rvol21 VG13-rvol39 VG13-rvol56 VG13-rvol8
VG01-BACKUP VG11-rvol1 VG12-rvol5 VG13-rvol22 VG13-rvol4 VG13-rvol57 VG13-rvol9
VG01-LOG VG11-rvol10 VG12-rvol6 VG13-rvol23 VG13-rvol40 VG13-rvol58 control
VG01-ORACLE VG11-rvol11 VG12-rvol7 VG13-rvol24 VG13-rvol41 VG13-rvol59 data1
VG01-WORK VG11-rvol12 VG12-rvol8 VG13-rvol25 VG13-rvol42 VG13-rvol6 data10
VG01-app VG11-rvol13 VG12-rvol9 VG13-rvol26 VG13-rvol43 VG13-rvol60 data11
VG01-log VG11-rvol2 VG13-rvol1 VG13-rvol27 VG13-rvol44 VG13-rvol61 data12
VG01-svc VG11-rvol3 VG13-rvol10 VG13-rvol28 VG13-rvol45 VG13-rvol62 data13
VG10-rvol1 VG11-rvol4 VG13-rvol11 VG13-rvol29 VG13-rvol46 VG13-rvol63 data14
VG10-rvol10 VG11-rvol5 VG13-rvol12 VG13-rvol3 VG13-rvol47 VG13-rvol64 data15
VG10-rvol11 VG11-rvol6 VG13-rvol13 VG13-rvol30 VG13-rvol48 VG13-rvol65 data16
VG10-rvol12 VG11-rvol7 VG13-rvol14 VG13-rvol31 VG13-rvol49 VG13-rvol66 data2
VG10-rvol2 VG11-rvol8 VG13-rvol15 VG13-rvol32 VG13-rvol5 VG13-rvol67 data3
VG10-rvol3 VG11-rvol9 VG13-rvol16 VG13-rvol33 VG13-rvol50 VG13-rvol68 data4
VG10-rvol4 VG12-rvol1 VG13-rvol17 VG13-rvol34 VG13-rvol51 VG13-rvol69 data5
VG10-rvol5 VG12-rvol10 VG13-rvol18 VG13-rvol35 VG13-rvol52 VG13-rvol7 data6
VG10-rvol6 VG12-rvol11 VG13-rvol19 VG13-rvol36 VG13-rvol53 VG13-rvol70 data7
```

```
VG10-rvol7  VG12-rvol2  VG13-rvol2  VG13-rvol37  VG13-rvol54  VG13-rvol71  data8
VG10-rvol8  VG12-rvol3  VG13-rvol20  VG13-rvol38  VG13-rvol55  VG13-rvol72  data9
```

5. rawdevice 할당

- > `/etc/sysconfig/rawdevices` 파일에 rawdevice 할당 추가
- > `/etc/udev/rules.d/60-raw.rule` 파일에 user, group, 퍼미션 설정 추가

```
=====
i=1
j=1
while (( $i < 51 ))
do
echo "/dev/raw/raw$i /dev/VG10/rvol$j" >> /etc/sysconfig/rawdevices
echo "ACTION==W"addW", KERNEL==W"raw$iW", OWNER=W"oraLBSW",
GROUP=W"dbaW", MODE=W"0640W"" >> /etc/udev/rules.d/60-raw.rules
let i+=1
let j+=1
done
=====
```

- > `/etc/sysconfig/rawdevices` 파일에 정상적으로 입력된 상태는 아래와 같다.

```
[root@localhost ~]#cat /etc/sysconfig/rawdevices
# raw device bindings
# format:  <rawdev> <major> <minor>
#          <rawdev> <blockdev>
# example: /dev/raw/raw1 /dev/sda1
#          /dev/raw/raw2 8 5
```

```
/dev/raw/raw1 /dev/VG10/rvol1
/dev/raw/raw2 /dev/VG10/rvol2
/dev/raw/raw3 /dev/VG10/rvol3
/dev/raw/raw4 /dev/VG10/rvol4
/dev/raw/raw5 /dev/VG10/rvol5
/dev/raw/raw6 /dev/VG10/rvol6
/dev/raw/raw7 /dev/VG10/rvol7
/dev/raw/raw8 /dev/VG10/rvol8
/dev/raw/raw9 /dev/VG10/rvol9
/dev/raw/raw10 /dev/VG10/rvol10
/dev/raw/raw11 /dev/VG10/rvol11
(이하 생략)
```

- > `/etc/udev/rules.d/60-raw.rules` 파일에 정상적으로 입력된 상태는 아래와 같다.

```
[root@localhost ~]#cat /etc/udev/rules.d/60-raw.rules
# Enter raw device bindings here.
```

```
#
# An example would be:
# ACTION=="add", KERNEL=="sda", RUN+="/bin/raw /dev/raw/raw1 %N"
# to bind /dev/raw/raw1 to /dev/sda, or
# ACTION=="add", ENV{MAJOR}=="8", ENV{MINOR}=="1", RUN+="/bin/raw
/dev/raw/raw2 %M %m"
# to bind /dev/raw/raw2 to the device with major 8, minor 1.
```

```
ACTION=="add", KERNEL=="raw1", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw2", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw3", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw4", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw5", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw6", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw7", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw8", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw9", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw10", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw11", OWNER="oraLBS", GROUP="dba", MODE="0640"
ACTION=="add", KERNEL=="raw12", OWNER="oraLBS", GROUP="dba", MODE="0640"
(이하 생략)
```

-> rawdevice 데몬 활성화 및 실행

```
[root@localhost rock]#chkconfig rawdevices on
[root@localhost rock]#service rawdevices start
```

-> rawdevice 데몬을 start 하면 아래와 같이 화면에 rawdevice 정보가 출력되고, /dev/raw 디렉토리에 생성된 rawdevice 파일을 확인할 수 있다.

```
[root@localhost ~]#/etc/init.d/rawdevices restart
Assigning devices:
```

```
    /dev/raw/raw1 --> /dev/VG10/rvol1
/dev/raw/raw1: bound to major 253, minor 19
    /dev/raw/raw2 --> /dev/VG10/rvol2
/dev/raw/raw2: bound to major 253, minor 20
    /dev/raw/raw3 --> /dev/VG10/rvol3
/dev/raw/raw3: bound to major 253, minor 21
    /dev/raw/raw4 --> /dev/VG10/rvol4
/dev/raw/raw4: bound to major 253, minor 22
    /dev/raw/raw5 --> /dev/VG10/rvol5
/dev/raw/raw5: bound to major 253, minor 23
(중간생략)
```

```
    /dev/raw/raw105 --> /dev/VG13/rvol69
/dev/raw/raw105: bound to major 253, minor 126
    /dev/raw/raw106 --> /dev/VG13/rvol70
/dev/raw/raw106: bound to major 253, minor 127
```

```
        /dev/raw/raw107 --> /dev/VG13/rvol71
/dev/raw/raw107:      bound to major 253, minor 128
        /dev/raw/raw108 --> /dev/VG13/rvol72
/dev/raw/raw108:      bound to major 253, minor 129
done
```

-> /dev/raw/ 디렉토리 하위에 rawdevice가 정상적으로 생성되었는지 확인한다.

```
[root@localhost raw]#ls -l
```

```
total 0
```

```
crw-r----- 1 oraLBS dba 162,  1 Aug 30 17:17 raw1
crw-r----- 1 oraLBS dba 162, 10 Aug 27 17:36 raw10
crw-r----- 1 oraLBS dba 162, 100 Aug 29 12:05 raw100
crw-r----- 1 oraLBS dba 162, 101 Aug 29 12:05 raw101
crw-r----- 1 oraLBS dba 162, 102 Aug 29 12:05 raw102
crw-r----- 1 oraLBS dba 162, 103 Aug 29 12:05 raw103
crw-r----- 1 oraLBS dba 162, 104 Aug 27 17:36 raw104
crw-r----- 1 oraLBS dba 162, 105 Aug 27 17:36 raw105
crw-r----- 1 oraLBS dba 162, 106 Aug 27 17:36 raw106
crw-r----- 1 oraLBS dba 162, 107 Aug 27 17:36 raw107
crw-r----- 1 oraLBS dba 162, 108 Aug 27 17:36 raw108
crw-r----- 1 oraLBS dba 162, 11 Aug 27 17:36 raw11
crw-r----- 1 oraLBS dba 162, 12 Aug 27 17:36 raw12
```

(중간생략)

```
crw-r----- 1 oraLBS dba 162, 92 Aug 27 17:36 raw92
crw-r----- 1 oraLBS dba 162, 93 Aug 27 17:36 raw93
crw-r----- 1 oraLBS dba 162, 94 Aug 29 12:05 raw94
crw-r----- 1 oraLBS dba 162, 95 Aug 29 12:05 raw95
crw-r----- 1 oraLBS dba 162, 96 Aug 29 12:05 raw96
crw-r----- 1 oraLBS dba 162, 97 Aug 29 12:05 raw97
crw-r----- 1 oraLBS dba 162, 98 Aug 29 12:05 raw98
crw-r----- 1 oraLBS dba 162, 99 Aug 29 12:05 raw99
```

```
[root@localhost raw]#
```