



DRBD Storage

DISTRIBUTE REPLICATION

강사소개

- 이름: 최 국현
- tang@linux.com

DRBD

배포 기반 블록 스토리지

2023-05-30

:0<-<

DRBD

DRBD는 정확히는 페이스메이커의 솔루션은 아니다. LINBIT회사에서 제작한 솔루션이며, 코어버전은 GPL 2.0으로 공개가 되어 있다.

DRBD 소스코드

DRBD는 RAID 1기능을 TCP기반으로 구현한다. Glusterfs와 기능이 비슷하지만, 파일 기반이 아닌 실제 물리적 블록 장치 기반으로 구현한다. 대다수 배포판에서 별도의 컴파일 없이 사용이 가능하도록 패키징이 되어 있기 때문에 바이너리 기반으로 설치 후, 설정하여 바로 사용이 가능하다.

DRBD는 GFS2파일 시스템과 다르게 BIT by BIT로 블록을 복제한다. 멀티 락킹 혹은 DLM지원하지 않는 파일 시스템도 안정적으로 사용이 가능하다.

보통 ACTIVE/ACTIVE 혹은 ACTIVE/PASSIVE형태로 많이 사용한다.

DRBD

DRBD는 정확히는 페이스메이커의 솔루션은 아니다. LINBIT회사에서 제작한 솔루션이며, 코어버전은 GPL 2.0으로 공개가 되어 있다.

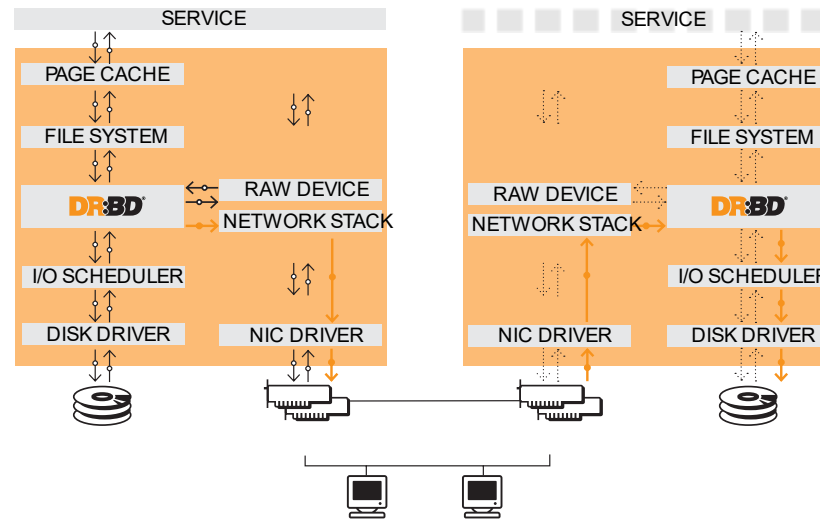
DRBD 소스코드

DRBD는 RAID 1기능을 TCP기반으로 구현한다. Glusterfs와 기능이 비슷하지만, 파일 기반이 아닌 실제 물리적 블록 장치 기반으로 구현한다. 대다수 배포판에서 별도의 컴파일 없이 사용이 가능하도록 패키징이 되어 있기 때문에 바이너리 기반으로 설치 후, 설정하여 바로 사용이 가능하다.

DRBD는 GFS2파일 시스템과 다르게 BIT by BIT로 블록을 복제한다. 멀티 락킹 혹은 DLM지원하지 않는 파일 시스템도 안정적으로 사용이 가능하다.

보통 ACTIVE/ACTIVE 혹은 ACTIVE/PASSIVE형태로 많이 사용한다.

DRBD



DRBD

DRBD is, by definition and as mandated by the Linux kernel architecture, agnostic of the layers above it. Therefore, it is impossible for DRBD to miraculously add features to upper layers that these do not possess. For example, DRBD cannot auto-detect file system corruption or add active-active clustering capability to file systems like ext3 or XFS.

DRBD

drbdadm

The high-level administration tool of the DRBD-utils program suite. Obtains all DRBD configuration parameters from the configuration file `/etc/drbd.conf` and acts as a front-end for `drbdsetup` and `drbdmeta`. `drbdadm` has a dry-run mode, invoked with the `-d` option, that shows which `drbdsetup` and `drbdmeta` calls `drbdadm` would issue without actually calling those commands.

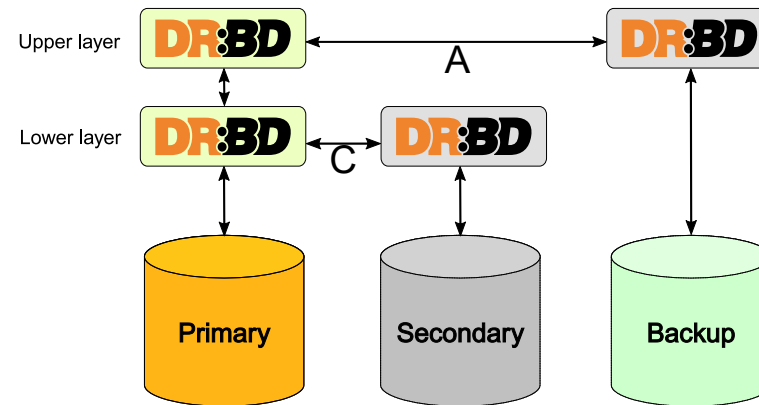
drbdsetup

Configures the DRBD module that was loaded into the kernel. All parameters to `drbdsetup` must be passed on the command line. The separation between `drbdadm` and `drbdsetup` allows for maximum flexibility. Most users will rarely need to use `drbdsetup` directly, if at all.

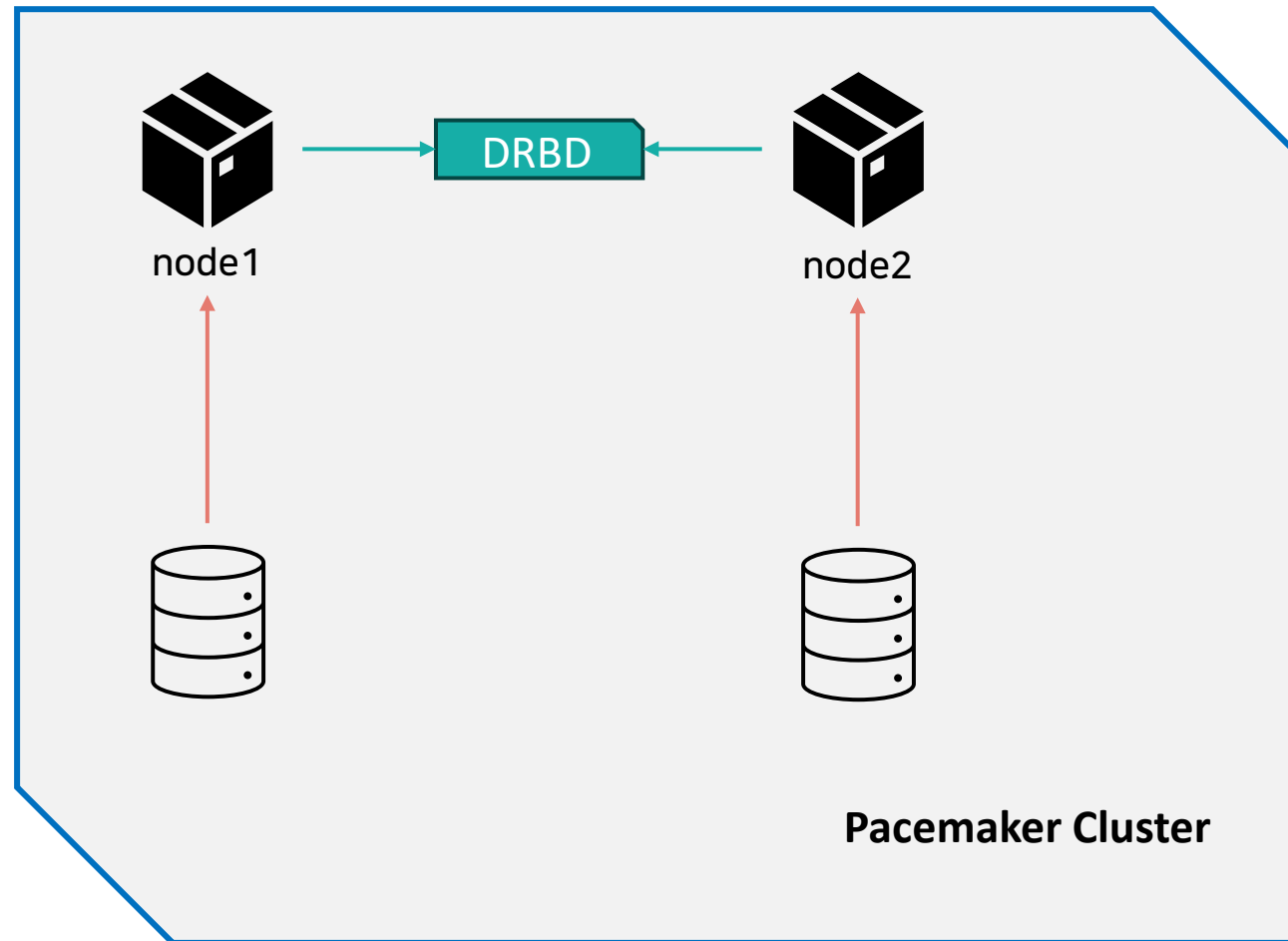
drbdmeta

Allows to create, dump, restore, and modify DRBD meta data structures. Like `drbdsetup`, most users will only rarely need to use `drbdmeta` directly.

DRBD



PACEMAKER



DRBD 호환성

Rocky 8/9, RHEL 8/9에서는 손쉽게 설치가 가능하지만, CentOS-8/9-Stream에서는 커널 호환성 문제로 drbd모듈이 올바르게 동작이 되지 않는다.

DRBD를 올바르게 사용하기 위해서는 가급적이면 **RHEL9/Alma/Rocky 9**에서 설치 및 운영을 권장한다. CentOS에서 설치 및 운영을 하기 위해서는 커널 버전을 RHEL 9과 동일하게 구성한다.

- DRBD는 각 리소스당 최대 32개 노드 접근 가능
- DRBD Volumes은 노드당 1048576개 연결 가능
- DRBD에서 최대 장치 크기는 1PiB(1024TiB)
- 리눅스 커널은 최소 3.10부터 지원

DRBD SETUP

Node1, Node2에 DRBD를 구성하여 xfs파일 시스템 기반으로 복제 및 DR를 구현한다. 다시 말하지만, **다중 읽기/쓰기** 기능은 지원하지 않는다. 동작 테스트 하기 위해서는 두 개의 노드를 이동하면서 사용한다.

```
node1/2# pvcreate /dev/sdb
```

```
node1/2# vgcreate drbd-demo /dev/sdb
```

```
node1/2# lvcreate --name drbd-demo -l 100%Free drbd-demo
```

```
node1/2# firewall-cmd --add-port=6996-7800/tcp --permanent
```

```
node1/2# firewall-cmd --reload
```

DRBD SETUP

```
# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source  
address="192.168.90.110" port port="7789" protocol="tcp" accept'
```

```
# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source  
address="192.168.90.120" port port="7789" protocol="tcp" accept'
```

```
# firewall-cmd --reload
```

```
# firewall-cmd --list-all
```

DRBD SETUP

```
node4# for i in {1..2} ; do ssh root@node${i} 'dnf install epel-release -y' ; done
```

```
node4# for i in {1..2} ; do ssh root@node${i} 'dnf install https://www.elrepo.org/elrepo-release-9.el9.elrepo.noarch.rpm -y' ; done
```

kernel-core-5.14.0-284.11.1.el9_2

```
node4# for i in {1..2} ; do ssh root@node${i} 'dnf update -y && reboot' ; done
```

```
node4# for i in {1..2} ; do ssh root@node${i} 'dnf install drbd drbd-bash-completion drbd-pacemaker drbd-utils kmod-drbd9x -y' ; done
```

```
node4# for i in {1..2} ; do ssh root@node${i} 'dnf install kernel -y && reboot' ; done
```

```
node4# for i in {1..2} ; do ssh root@node${i} 'depmod -a && modprobe drbd' ; done
```

```
node4# for i in {1..2} ; do ssh root@node${i} 'systemctl enable --now drbd' ; done
```

```
node4# for i in {1..2} ; do ssh root@node${i} 'systemctl status drbd' ; done
```

주의: uEFI가 활성화 되어 있으면, 모듈이 올바르게 메모리에 상주가 안됩니다. 끄고 하세요!

```
# modprobe drbd  
modprobe: ERROR: could not insert 'drbd': Key was rejected by service
```

DRBD SETUP

```
cat<< EOF> /etc/drbd.d/resource0.res
```

```
resource resource0 {
```

```
on node1.example.com {
```

```
device /dev/drbd1;
```

```
disk /dev/drbd-demo/drbd-demo;
```

```
address <IP_ADDRESS>:7789;
```

```
meta-disk internal;
```

```
}
```

```
on node2.example.com {
```

```
device /dev/drbd1;
```

```
disk /dev/drbd-demo/drbd-demo;
```

```
address <IP_ADDRESS>:7789;
```

```
meta-disk internal;
```

```
}
```

```
}
```

```
EOF
```

DRBD SETUP

```
node1# drbdadm create-md resource0
```

```
node1# drbdadm up resource0
```

```
node1# drbdadm status resource0
```

```
node1# drbdadm primary --force resource0
```

```
node1# drbdadm status resource0
```

```
node1# lsblk
```

```
node1# mkfs.xfs /dev/drbd1 or mkfs.ext4 /dev/drbd1
```

```
node1# mkdir -p /mnt/drbd
```


DRBD SETUP

```
node1# mount /dev/drbd1 /mnt/drbd
```

```
node1# cd /mnt/drbd
```

```
node1# touch test{1..100}
```

```
node1# umount /mnt/drbd
```

```
node1# drbdadm secondary resource0
```

```
node2# drbdadm primary resource0
```

```
node2# mkdir -p /mnt/drbd
```

```
node2# mount /dev/drbd1 /mnt/drbd
```

```
node2# ls -l /mnt/drbd
```