16-4. RAID

RAID

RAID의 필요성

- 서버의 사용자가 서버 운영에 있어 가장 당혹스러울 때는 하드디스크의 장애로 인한 DATA 손실이다.
- 이로 인해 백업이 절대적으로 필요한 경우가 있고 또한, 여분의 디스크가 있어 용량을 증설하려고 할 때 데이터 손실 없이 증설이 필요한 경우가 있다. 그래서 많은 서버 관리자는 RAID 구성을 통해 하드디스크의 가용성을 높이거나 서버 데이터의 안정성을 확보해야 한다.

RAID란?

- Redundant Array of Inexpensive Disks
    - SLED(Single Large Expensive Disk): RAID의 반대 개념
- 여러 개의 디스크를 배열하여 속도의 증대, 안정성의 증대, 효율성, 가용성의 증대를 하는데 쓰이는 기술이다.
- **장점**
    - 운용 가용성, 데이터 안정성 증대
    - 디스크 용량 증설의 용이성
    - 디스크 I/O 성능 향상

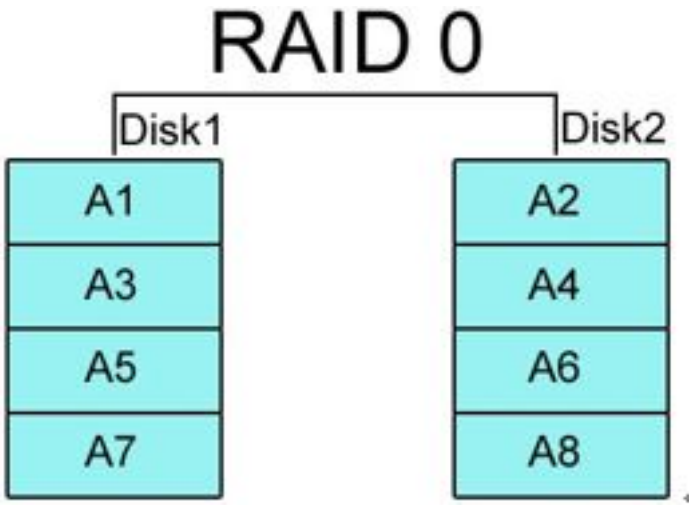
RAID의 종류와 구성방식

RAID 0

- RAID 0에는 Concatenate 방식과 Stripe 방식 두 가지 방식이 있다.
- **Concatenate 방식**(Span 볼륨)
    - 두개 이상의 디스크에 데이터를 순차적으로 쓰는 방법

- 장점: 디스크 기본 공간이 부족할 때 데이터는 보존하며 여분의 디스크를 볼륨에 포함하여 용량 증설이 가능하다.
- 단점: RAID 0의 특성상 디스크 중 하나의 디스크라도 장애가 발생하면 복구가 어렵고, 패리티(오류검출기능)를 지원하지 않는다.
- 용량: 모든 디스크의 용량을 합친 용량 (300GB disk * 2ea = 600GB)

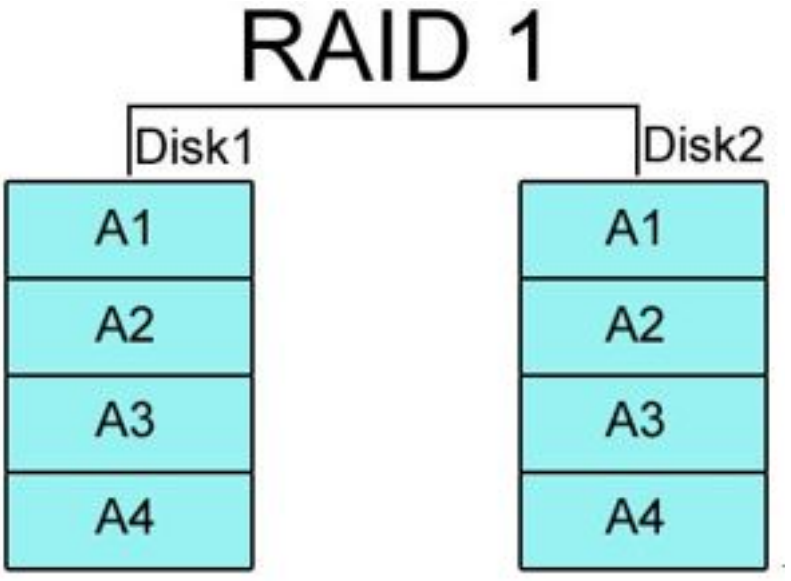
- **Stripe 방식**=흔히 RAID 0라고 하면 Stripe 방식



  - 두개 이상의 디스크에 데이터를 랜덤하게 쓰는 방법이다.
  - 장점: 데이터를 사용할 때 I/O를 디스크 수만큼 분할하여 쓰기 때문에 I/O 속도가 향상되고 I/O Controller나 I/O board 등 I/O를 담당하는 장치가 별도로 장착된 경우 더 큰 I/O 속도 향상 효과를 볼 수 있다.
  - 단점: Stripe를 구성할 시 기존 데이터는 모두 삭제 돼야 한다. 그외의 단점은 위의 Concat 방식과 같다.

RAID 1(Mirror)

- Mirror 볼륨 내의 패리티를 사용하지 않고 디스크에 같은 데이터를 중복 기록하여 데이터를 보존하게 되며, 적어도 동일한 용량의 디스크 두 개가 필요하다.

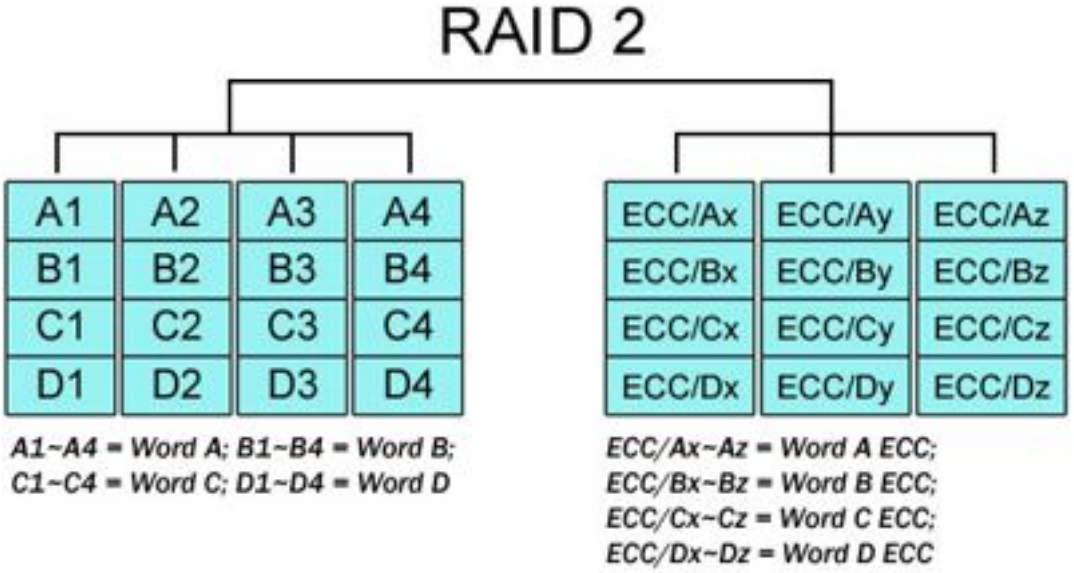


- 장점: 볼륨 내 디스크 중 하나의 디스크만 정상이어도 데이터는 보존되어 운영이 가능하기 때문에 가용성이 높고, 복원이 비교적 매우 간단하다.

- 단점: 용량이 절반으로 줄고, 쓰기 속도가 조금 느려진다.
- 용량: 모든 디스크의 절반의 용량 (300GB *2ea = 300GB)
    - 실제 사용량=전체용량 x 1/2

RAID 2

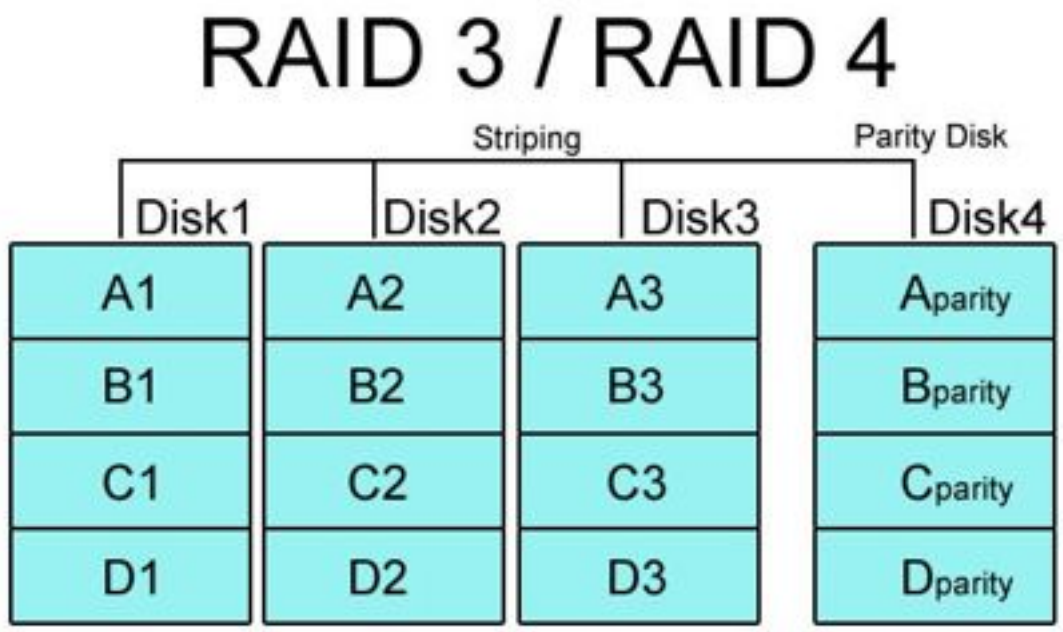
- RAID 0처럼 striping 방식이지만 에러 체크와 수정을 할 수 있도록 Hamming code를 사용하고 있는 것이 특징이다.
- 하드 디스크에서 ECC(Error Correction Code)를 지원하지 않기 때문에 ECC를 별도의 드라이브에 저장하는 방식으로 처리된다.



- 하지만 ECC를 위한 드라이브가 손상될 경우는 문제가 발생할 수 있으며 패리티 정보를 하나의 하드 드라이브에 저장하는 RAID 4가 나오면서 거의 사용되지 않는 방식이다.

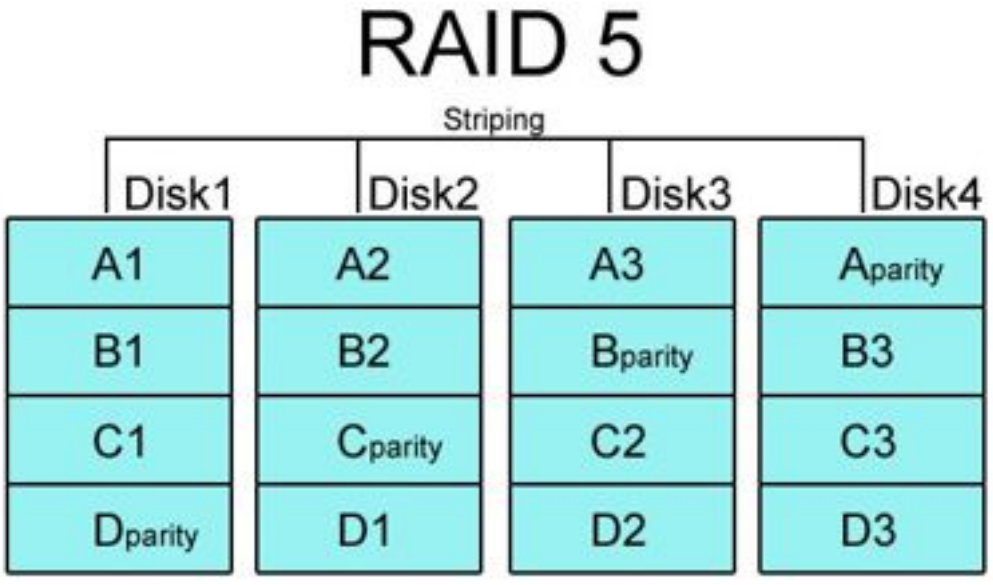
RAID 3, RAID 4

- RAID 3, RAID 4는 RAID 0, RAID 1의 문제점을 보완하기 위한 방식으로 3, 4로 나뉘긴 하지만 RAID 구성 방식은 거의 같다.
- RAID 3, 4는 기본적으로 RAID 0과 같은 striping 구성을 하고 있어 성능을 보완하고 디스크 용량을 온전히 사용할 수 있게 해주는데 여기에 추가로 에러 체크 및 수정을 위해서 패리티 정보를 별도의 디스크에 따로 저장한다.
- RAID 3은 데이터를 바이트 단위로 나누어 디스크에 동등하게 분산 기록하며 RAID 4는 데이터를 블록 단위로 나눠 기록하므로 완벽하게 동일하진 않다는 차이가 있다.
- RAID 3은 드라이브 동기화가 필수적이라 많이 사용되지 않고 RAID 4를 더 많이 쓴다고 보면 된다.



RAID 5

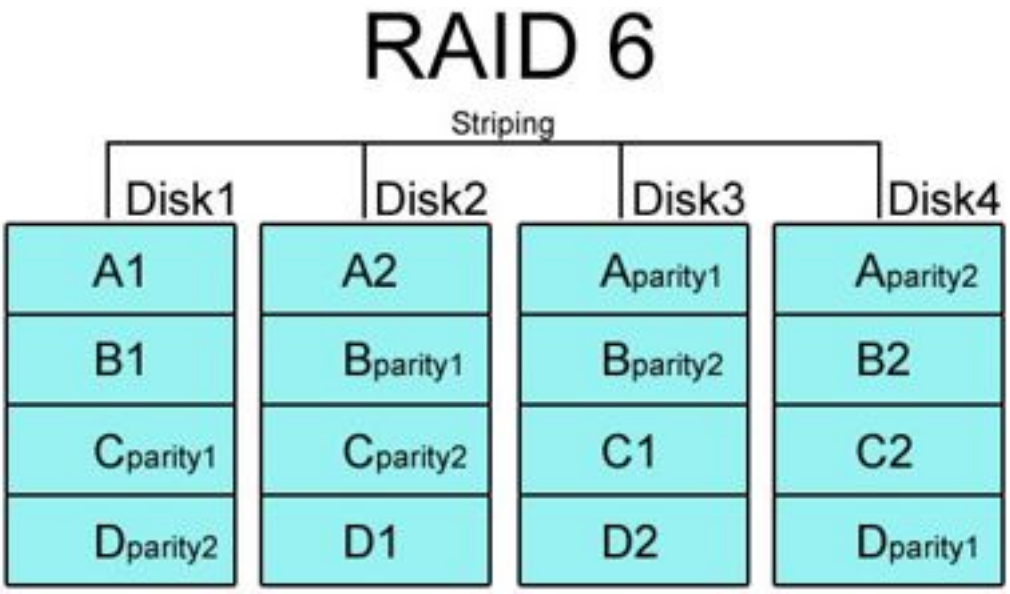
- RAID 3,4에서 별도의 패리티 정보 디스크를 사용함으로써 발생하는 문제점을 보완하는 방식으로 패리티 정보를 stripe로 구성된 디스크 내에서 처리하게 만들었다.
- 만약 1개의 하드가 고장나더라도 남은 하드들을 통해 데이터를 복구할 수 있다는 장점이 있다.
- 용량
    - 실제 사용량 = N-1



RAID 6

- RAID 6은 RAID 5와 같은 개념이지만 다른 드라이브들 간에 분포되어 있는 2차 패리티 정보를 넣어 2개의 하드에 문제가 생겨도 복구할 수 있게 설계되었으므로 RAID 5보다 더욱 데이터의 안전성을 고려하는 시스템에서 사용된다.
- 용량

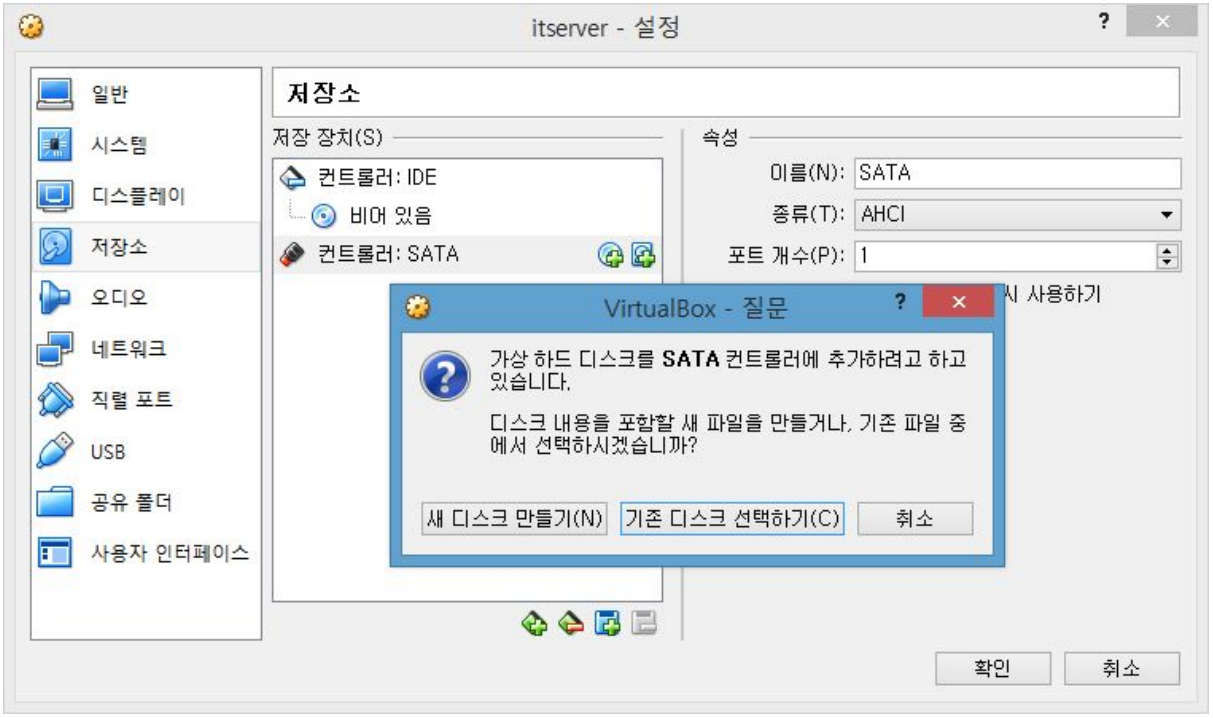
- 실제 사용량 = N - 2



RAID 구현

준비 1단계: 가상 머신에서 디스크 추가하기

- itserver를 종료시킨다.
- Oracle VM Virtual Box 관리자에서 **itserver**를 선택한 후 <설정> 버튼을 클릭한다.
  - 왼쪽의 [저장소]를 클릭하고, '컨트롤러: SATA'를 선택하고 두번째 아이콘인 <하드 디스크를 추가합니다>를 클릭
  - 메시지가 나오면 <새 디스크 만들기>를 클릭



- [하드 디스크 파일 종류]에서 기본인 'VDI'를 선택한 상태에서 <다음>을 클릭
- [물리적 하드 드라이브에 저장]에서 기본인 '동적 할당'이 선택된 상태에서 <다음>을 클릭
- [파일 위치 및 크기]에서 경로를 미리 만들어둔 'D:\centos6\server'로 맞춘 후, 크기를 1G로 변경하고 <만들기>를 클릭하여 가상 디스크를 생성
- 위와 같이 **모두 9개(마지막은 2G)**를 추가 생성

준비 2단계: 디스크 파티션 설정하기

- **fdisk**

기능	디스크의 파티션 생성, 삭제, 보기 등 파티션을 관리한다.
형식	**fdisk 옵션 장치명**
옵션	-b 크기: 섹터의 크기를 지정 -l: 파티션 테이블을 표시
사용 예	fdisk /dev/sdb fdisk -l

- fdisk -l
  - 전체 디스크의 파티션 정보를 확인한다.
- fdisk /dev/sdc
  - fdisk 장치명을 통해 명령을 실행한다.
  - n: 새로운 파티션을 생성
  - p: 파티션 종류를 선택
  - 1: 파티션 번호를 선택
  - 시작 섹터: 기본값을 사용
  - 마지막 섹터: 기본값을 사용: 하드디스크 전체를 하나의 파티션 지정t
  - t: 파일 시스템 유형 선택
  - **fd: Hex Code를 Linux raid autodetect** 유형번호를 입력
    - L을 입력하면 전체 유형이 출력
  - p: 설정 내용을 확인
  - w: 파티션 설정 정보를 디스크에 기록하고 fdisk를 종료
- fdisk -l
  - 설정한 파티션이 제대로 되었는지 확인한다.
- 위와 같이 총 9개를 파티션 지정한다.

RAID 0 구축

0단계: RAID 구성

- **mdadm**

기능	RAID를 생성, 관리한다.
형식	**mdadm --create 이름 --level=레벨 --raid-devices 갯수 물리적파티션 ...**

옵션	--stop 이름: RAID 장치 중지 --run 이름: RAID 장치 가동 --detail 이름: RAID 장치 내용 출력
사용 예	mdam --create /dev/md0 --level=0 --raid-devices=2 /dev/sdc1 /dev/sdd1 mdadm --detail /dev/md0 mdadm --stop /dev/md0 mdadm --run /dev/md0

- **mdadm --detail --scan**
  - 확인 명령

1단계: 파일 시스템 생성

- mkfs -t ext4 /dev/md0
- mkfs.ext4 /dev/md0

2단계: 마운트

- mkdir /raid0
- mount /dev/md0 /raid0

3단계: fstab 설정

- vi /etc/fstab
  - /dev/md0 /raid0 ext4 defaults 1 2

RAID 1 구축

0단계: RAID 구성

- **mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sde1 /dev/sdf1**
  - **Continue creating array? 메시지 나오면 y를 입력하여 계속 진행**
- **mdadm --detail --scan**
  - 확인 명령

1단계: 파일 시스템 생성

- mkfs -t ext4 /dev/md1
- mkfs.ext4 /dev/md1

2단계: 마운트

- mkdir /raid1
- mount /dev/md1 /raid1

3단계: fstab 설정

- vi /etc/fstab
  - /dev/md1 /raid1 ext4 defaults 1 2

RAID 5 구축

0단계: RAID 구성

- **mdadm --create /dev/md5 --level=5 --raid-devices=3 /dev/sdg1 /dev/sdh1 /dev/sdi1**
- **mdadm --detail --scan**
  - 확인 명령

1단계: 파일 시스템 생성

- mkfs -t ext4 /dev/md5
- mkfs.ext4 /dev/md5

2단계: 마운트

- mkdir /raid5
- mount /dev/md5 /raid5

3단계: fstab 설정

- vi /etc/fstab
    - /dev/md5 /raid5 ext4 defaults 1 2

Linear RAID 구축

0단계: RAID 구성

- **mdadm --create /dev/md9 --level=linear --raid-devices=2 /dev/sdj1 /dev/sdk1**
- **mdadm --detail --scan**
    - 확인 명령

1단계: 파일 시스템 생성

- mkfs -t ext4 /dev/md9
- mkfs.ext4 /dev/md9

2단계: 마운트

- mkdir /raid9
- mount /dev/md9 /raid9

3단계: fstab 설정

- vi /etc/fstab
    - /dev/md9 /raid9 ext4 defaults 1 2