

# 파일백업을위한 bareos 소개

## 시작하는 말

안녕하세요, 고니 입니다.

기존에 작성했던 콘텐츠를 업데이트를 하면서 문서의 리팩토링(Refactoring)을 진행해보려고 합니다.

IT인프라를 운영하다보면 시스템이 안정적으로 운영되는건 물론이고, 백업에 대한 중요성도 증가하게 되죠.

랜섬웨어(Ransomware)에 걸렸을 경우, 조치할 수 있는 방법중에 하나가 백업된 내용을 복구하는것도 방법이 될꺼고요.

누군가 그러더라구요. IT인프라 운영의 마지막은 백업이라고....

백업 방식에 대해 정리해볼께요.

## 백업방식

### 1. Full Backup (전체백업)

- 백업되는 시점의 모든 데이터를 백업하는 방식입니다.
- 데이터양이 많아지면 백업되는 시간이 증가되겠지만, 복구되는 시점은 가장 간단한 방법으로 복구를 할 수 있습니다.

### 2. Incremental Backup (증분백업)

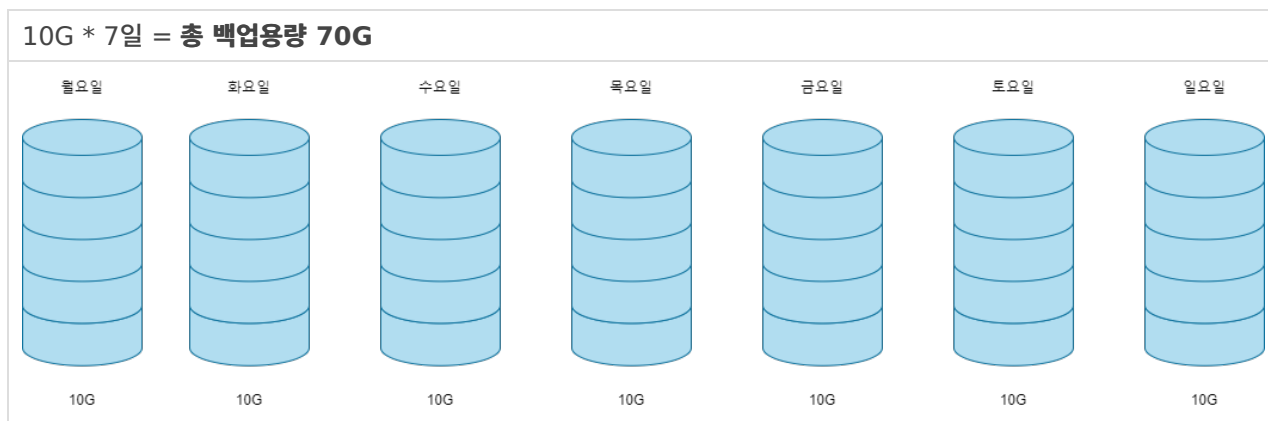
- 지난 백업 이후 변경된 사항에 대해서만 백업을 수행합니다.
- 변경된 데이터만 백업하기 때문에 Full backup보다 백업의 소요시간이 단축되죠, 복구절차의 경우 Full backup 이후 변경된 데이터만 순차적으로 복구해야하기 때문에 Full Backup보다는 백업절차가 증가될수 있습니다.

### 3. Decremental Backup (차등백업)

- 지난 백업 이후 변경된 사항을 포함한 백업을 수행합니다.
- 변경된 모든 데이터를 포함하고 있기 때문에 증분백업보다는 백업속도와 용량이 증가할 수 있지만, 복구때는 전체 백업 과 백업데이터만 보관하고 있으면 되기 때문에 증분백업보다는 백업절차가 간소화 될 수 있습니다

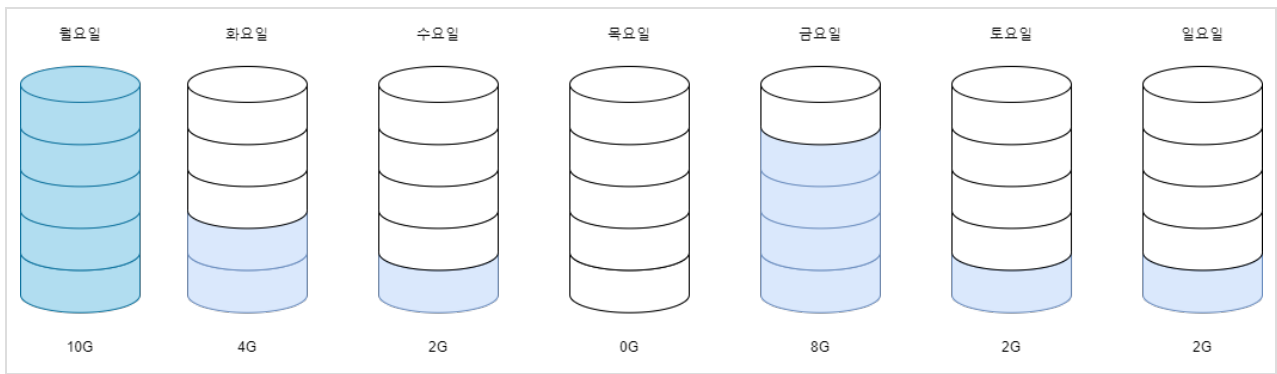
### 4. 백업방법별 용량 비교

- Full backup

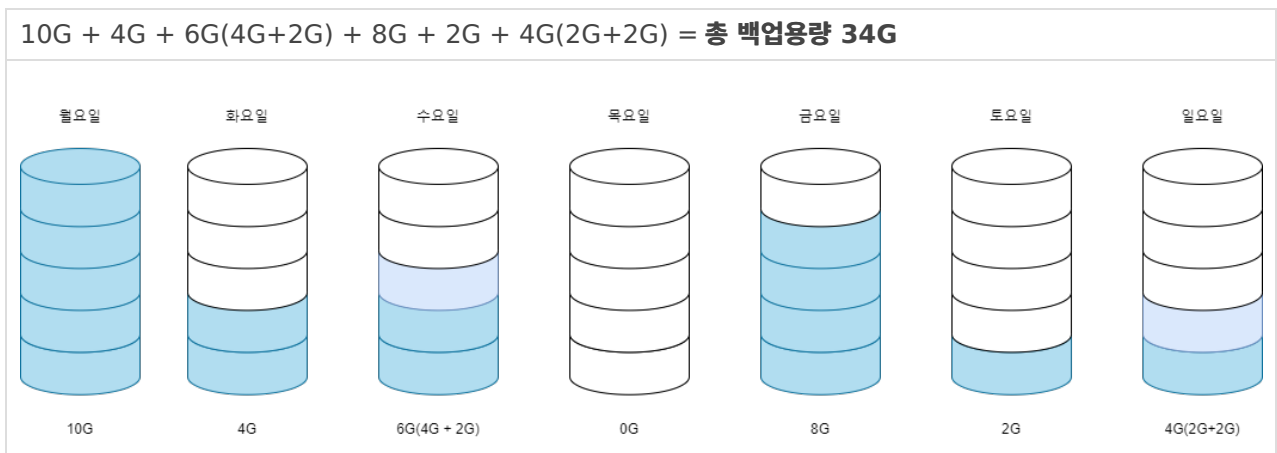


- Incremental Backup

10G + 4G + 2G + 0G + 8G + 2G + 2G = 총 백업용량 28G



- Dcremental Backup



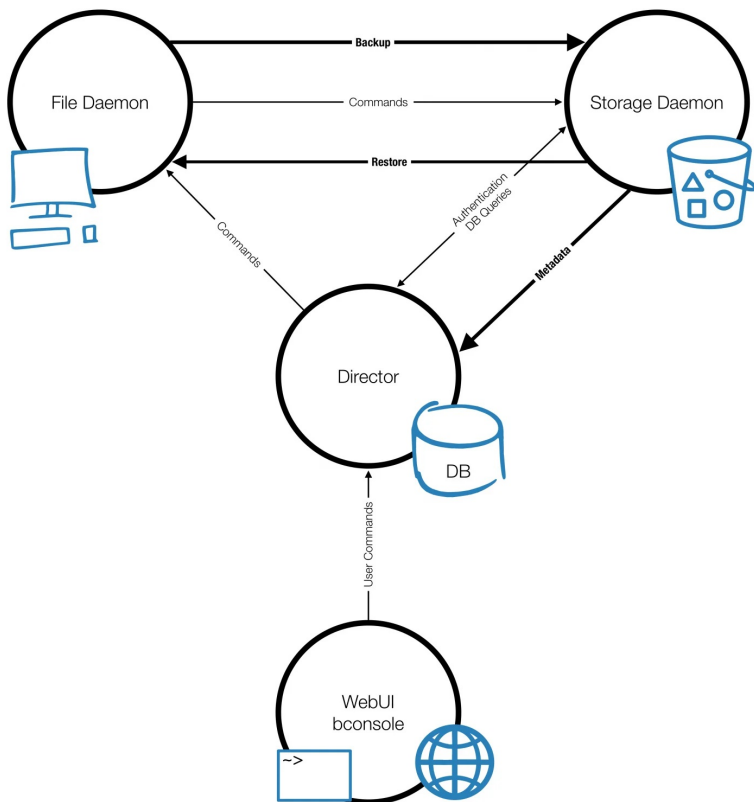
이제 약장수가 약 팔듯이, 제가 운영했던 Opensource Backup 솔루션을 소개해보려고 합니다.

왜냐하면, Backup솔루션 자체가 굉장히 고가의 솔루션이기도 하고, 상용솔루션 만큼 백업과 복구가 가능했거든요.

## Bareos 소개

1. 2000년 1월에 출시한 Opensource Backup인 Bacula의 Fork된 OSS Backup솔루션
2. Bacula는 AGPL v3 + 독점라이선스 , Bareos는 AGPL v3 로 운영([커뮤니티 vs 서브스크립션](#))
3. Bacula는 독점라이선스로 운영하면서 커뮤니티보다는 B2B로 모델을 중점으로 두면서 2011년에 Bareos로 Forking되었음
4. Linux(Centos/Ubuntu/Suse/Debian), Windows등 다양한 OS 환경에서 설치가능
5. '19년 6월 기준 최신 릴리즈는 18.2.5 버전이 릴리즈.

## Bareos 기본 시스템 구성도



1. Director : 백업 정책 / 스케줄 / 클라이언트를 관리하는 모듈
2. sd(storage-daemon) : 백업데이터를 저장하는 모듈
3. fd(file-daemon) : 백업할 데이터가 저장된 클라이언트
4. bconsole : 콘솔용 프로그램 (클라이언트 생성 및 백업 정책 설정은 불가)
5. webui : 웹기반에서 백업 / 복구 job을 수행할 수 있는 모듈 (클라이언트 생성 및 백업 정책 설정은 불가)

# Bareos설치하기

## 1. Director 설치하기

- Director을 위한 패키지 설치

```
$> wget https://download.bareos.org/current/EL_7/bareos.repo /etc/yum.repos.d/bareos.repo
$> yum install bareos bareos-director bareos-common bareos-database-mysql -y
```

- DB구성 (순서대로 진행)

```
$> /usr/lib/bareos/scripts/create_bareos_database mysql
$> /usr/lib/bareos/scripts/make_bareos_tables mysql
$> /usr/lib/bareos/scripts/grant_bareos_privileges mysql
```

- DB구성 파일 생성

```
$> cat /etc/bareos/bareos-dir.d/catalog/MyCatalog.conf
Catalog {
  Name = MyCatalog
  dbdriver = "mysql"
  dbname = "{{ DB이름 }}"
  dbuser = "{{ DB접속계정 }}"
  dbpassword = "{{ DB접속패스워드 }}"
  dbsocket=/tmp/mysql.sock
}
```

- Director → SD구성을 위한 파일 정보 수정

```
$> cat /etc/bareos/bareos-dir.d/storage/File.conf
Storage {
```

```

Name = File
Address = {{ SD서버IP }}
Password = "{{ SD연동용 패스워드 }}" // SD서버용 패스워드 설정과 동일해야 함
Device = FileStorage1
Device = FileStorage2
Maximum Concurrent Jobs = 2
Media Type = File
}

```

## • 스케줄 정보

#백업 수행을 위한 설정 (기본값은 모듈별로 분리되어 있으나, 운영상 편리를 위해 통합된 config가 효율적인듯...)

```

파일명 : /etc/bareos/bareos-dir.d/job/{{클라이언트 이름}}.conf
$> cat /etc/bareos/bareos-dir.d/schedule/{{백업 정책 이름}}.conf
Schedule {
  Name = "{{ 스케줄 이름 }}"
  Run = Full  mon at 01:00      #Full 백업 수행 날짜와 시간 수행
  Run = Incremental tue-sun at 01:00  #Incremental 백업 수행 날짜와 시간 수행
}

```

## • 클라이언트 정보

```

Client {
  Name = {{ 클라이언트 이름 }}
  Description = "Client resource of the Director itself."
  Address = localhost
  Password = "{{ 클라이언트접속용 패스워드 }}"      # password for FileDaemon
}

```

## • 파일셋 설정

```

FileSet {
  Name = "{{ 클라이언트 이름 }}"
  Include {
    Options {
      compression=lz4  #파일 압축방식
      signature=SHA1  #암호화 방식 (sha1 / md5 중 가능)
      Sparse = yes
    }
    File = "/usr/sbin" #백업 경로
    File = "/etc/bareos/"
    File = "/etc/bareos-webui/"
  }
}

```

## • 백업 Job정보

```

$>
Job {
  Name = "{{ 클라이언트 이름 }}"
  JobDefs = "{{ 클라이언트 이름 }}"
  Client = "{{ 클라이언트 이름 }}"
}

```

## • 백업 정책 정보

```

$>
JobDefs {
  Name = "{{ 클라이언트 이름 }}"
  Type = Backup
  Level = {{ 백업 정책이름 }}
  Client = {{ 클라이언트 이름 }}
  FileSet = {{ 클라이언트 이름 }}
  Schedule = "{{ 스케줄이름 }}"
  Storage = File
}

```

```

Messages = Standard
Pool = {{ 클라이언트 이름 }}
Priority = 10
Write Bootstrap = "/var/lib/bareos/%c.bsr"
}

```

- 백업 Pool 정보

```

$>
Pool {
  Name = {{ 클라이언트 이름 }}
  Pool Type = Backup
  Recycle = yes                # 볼륨 재사용 여부
  AutoPrune = yes              # 보관기간 만료시 Pool에 write되지 않도록 정책 설정
  Volume Retention = 30 days    # 볼륨 보관 기간
  Maximum Volume Bytes = 100G    # 파일당 최대기록 용량
  Maximum Volumes = 100         # 볼륨당 최대 생성 갯수
  Label Format = "{{ 클라이언트 이름 }}-" # Pool생성 이름
  Purge Oldest Volume = yes
}

```

## 2. SD구성

- 패키지 설치

```
$> yum install bareos-storage -y
```

- SD실행을 위한 설정

```

$> cat /etc/bareos/bareos-sd.d/storage/bareos-sd.conf
Storage {
  Name = {{ SD이름 }}
  Maximum Concurrent Jobs = 20
}

```

- Director와 연동을 위한 설정

```

$> cat /etc/bareos/bareos-sd.d/director/bareos-dir.conf
Director {
  Name = {{ Director 이름 }}
  Password = "{{ SD연동용 패스워드 }}"
}

```

- 스토리지 구성

```

$> cat /etc/bareos/bareos-sd.d/device/FileStorage.conf

Device {
  Name = FileStorage1
  Media Type = File
  Archive Device = /backup <-- 백업 데이터 저장 경로
  LabelMedia = yes;           # lets Bareos label unlabeled media
  Random Access = yes; <-- HDD처럼 랜덤엑세스가 가능한 매체인 경우 yes로 변경(tape는 no로 설정)
  AutomaticMount = yes;
  RemovableMedia = no;
  AlwaysOpen = yes;
  Maximum Concurrent Jobs = 10
}

```

## 3. FD설정

1. 패키지 설치

```
$> yum install bareos-filedaemon bareos-fd -y
```

## 2. FD서비스 설정

```
$> cat /etc/bareos/bareos-fd.d/client/myself.conf
Client {
  Name = {{클라이언트 이름 }}
  Maximum Concurrent Jobs = 20
}
```

## 3. Director 연동

```
$> cat /etc/bareos/bareos-fd.d/director/bareos-dir.conf
Director {
  Name = {{ Director 이름 }}
  Password = "{{ Director 접속 패스워드 }}"
}
```

# 백업 작업 수행

1. 백업 / 복구 작업은 webui / bconsole을 이용해 수행이 가능하나, 운영상 편리를 위해 webui에서 수행
2. Webui 접근 후 Jobs → Run 메뉴 이동
  1. Job : 백업 작업 선택
  2. Client : 백업 클라이언트이름
  3. Fileset : 백업 대상
  4. Storage : File (기본값)
  5. Pool : 백업데이터 보관주기 설정
  6. Level : 백업유형 (Full / Incremental 중 선택)
  7. Priority : 백업 작업 우선순위 선택
  8. When : 백업 수행시간 (옵션)
3. Jobs → Show 메뉴 이동
4. 백업 결과 확인

# 복구 작업 수행

1. Restore 메뉴 이동
  - Client : 복구수행할 클라이언트
  - Backup Jobs : 백업 했던 job
  - Merge all client fileset : 여러개의 파일셋으로 백업된 경우 하나로 합쳐서 복구 수행 (단일 파일셋으로 구성하는 경우 영향없음)
  - Merge all related jobs to last full backup of selected backup job : 최근에 수행한 백업데이터로 복구 수행
  - Restore to client : 복구할 클라이언트 수행
  - Restore job : 기본값
  - replace files on client : 복구 대상에에서 파일 저장 정책
    1. always : 복구파일로 덮어쓰
    2. never : 기존파일 보관
    3. if file being restored is older than existing file : 백구 파일이 오래된 경우 복구파일로 덮어쓰
    4. if file being restored is newer than existing file : 백구 파일이 최신인 경우 복구파일로 덮어쓰
  - Restore location on client : 복구 파일 경로
2. 복구 작업 수행결과 확인

## Appendix#1. 커뮤니티 / 서브스크립션 비교

|                         | Community Edition   | Bareos Subscription  |
|-------------------------|---|--|
|                         | Packages  |  |
| Previous major releases | (Source code only)  | Subscription repository                                    |
| Current major release   | Community repository  | Subscription repository                                    |
| Development binaries    | <a href="https://download.bareos.org/next/">https://download.bareos.org/next/</a>               |  |
|                         | Community   |  |
| Mailing list            | <a href="#">bareos-users</a>  |  |
|                         | Support   |  |
| Basis                   | —   | 3 incidents, service level: not guaranteed                 |
| Standard                | —   | 10 incidents, service level: 2 business days               |
| Advanced                | —   | 20 incidents, service level: next business days            |
| Premier                 | —   | 50 incidents, service level: 4 h during business hours     |
| Global                  | —   | Unlimited number of incidents, response time: 1 hour, 24/7 |
|                         | Development   |  |
| Source code             | <a href="#">GitHub</a>  |  |
| Issue tracker           | <a href="https://github.com/bareos/bareos/issues/">https://github.com/bareos/bareos/issues/</a> |  |
| New features            | <a href="#">Join in</a>   | Co-Funding   |

# Reference

- [Bareos 공식사이트](#)
- [Bareos Github](#)