Backup & Emergency

gony

Contents

Backup

- Backup이란?
- Backup의 중요성
- Backup의 종류
  - Full, Incremental
  - Differential, Mirror
- Backup을 위한 파일 묶기/압축
  - 파일 아카이브 (tar, gz)
- Backup 하기
  - dd, rsync
  - 주기적인 백업 (cron)

Emergency

- 요인
  - Software, Hardware, Human
- 파일 시스템 (Software) 복구
  - fsck
- 여러 Emergency
  - 정전, 해킹, 서버실 이슈
- AWS에서 안전하게 사용하기
  - S3, Glacier, RDS

Convenience

- IaC
  - terraform

Backup

Backup이란?

- 잘못된 조작에 의한 데이터 파일 등의 소실에 대비하여 카피를 만들어 놓는 것
- Ex) 사진, 카카오톡 내용, 코드 등

- 여러 원인
  - 실수
  - 하드웨어/소프트웨어 문제
  - 도난
  - 바이러스 및 테러

Backup이 왜 필요한가?

- 예시

```
gony@DESKTOP-DN947S7:~/cron$ ls
date.sh  date.txt  math.sh  math.txt
gony@DESKTOP-DN947S7:~/cron$ rm math.sh
```

디버깅 중에

rm math.txt를 해야 하는데, 핵심 코드인 rm math.sh를 해버렸다?

news.joins.com › article

불법 아이템 복제 '리니지'…결국 22일로 서버 '롤백' - 중앙일보

2014. 1. 29. — 불법 아이템 복제 '리니지'…결국 22일로 서버 '롤백' ... 월 24일을 기점으로 해당 현상을 악용하는 사례가 급격히 확산되는 양상을 보였다"고 밝혔다.

이현 ▸ 생활코딩

2020. 10. 12. · ...커뮤니티에서 글을 보던도중 sudo rm -rf /*이라는 명령어가 brew 인스톨러와 잔여캐시를 제거해준다는 글을 보고 해당 명령어를 실행해버렸습니다. 실행하고 찾아보니 시스템까지 포맷하는 명령어 더라구요 현재 당황해서 급하게 터미널 종료후 강제 종료했습니다 이걸로 막아질까요? 정말 절실합니다. 만약 다 지워진다면 복구 가능할까요?

159 댓글 46개

Backup의 종류

- Full backup
- Incremental backup
- Differential backup
- Mirror backup

Backup의 종류: Full Backup

- 백업을 위해 선택한 모든 파일과 폴더를 백업하는 방법.
- 초기 또는 첫 번째 백업으로 일반적으로 사용
- 모든 파일과 폴더를 백업
  > 시간이 오래 걸림, 저장 공간 많이 차지
- 복구하기 간편
- 다른 종류의 백업에 영향을 줄 수 있기 때문에 주기적인 관리 필요

Backup의 종류: Full Backup

백업 방식	요일	백업 분량	복원 시 필요 파일
전체 Full	월	2G	월요일자 백업 파일 1개
〃	화	3G	화요일자 백업 파일 1개
〃	수	4G	수요일자 백업 파일 1개
〃	목	5G	목요일자 백업 파일 1개
〃	금	6G	금요일자 백업 파일 1개
〃	토	7G	토요일자 백업 파일 1개
〃	일	8G	일요일자 백업 파일 1개
〃	월	9G	월요일자 백업 파일 1개
특징	매 백업마다 많은 시간과 자원이 필요하다.		

Backup의 종류: Incremental Backup

- 마지막 백업 이후 수정된 내용 백업

  > Full backup에 의존적

- 적은 내용 > 백업 속도가 빠르다

- 복구 할 때 여러 파일을 각각 백업하므로, 복원 시간이 오래 걸린다.

Backup의 종류: Incremental Backup

백업 방식	요일	백업 분량	복원 시 필요 파일
전체 Full	월	2G	월요일자 백업 파일 1개
증분 Incremental	화	1G	월요일자 백업 파일 1개와 화요일자 증분 백업 파일 1개
〃	수	1G	월요일자 백업 파일 1개와 화~수요일자 증분 백업 파일 2개
〃	목	1G	월요일자 백업 파일 1개와 화~목요일자 증분 백업 파일 3개
〃	금	1G	월요일자 백업 파일 1개와 화~금요일자 증분 백업 파일 4개
〃	토	1G	월요일자 백업 파일 1개와 화~토요일자 증분 백업 파일 5개
〃	일	1G	월요일자 백업 파일 1개와 화~일요일자 증분 백업 파일 6개
전체	월	9G	월요일자 백업 파일 1개
특징	이전 백업 파일 모두를 순서대로 복원해야 한다.		

Backup의 종류: Differential Backup

- 마지막 full backup이후 변경된 모든 내용을 백업
- 백업 속도는 incremental backup보다는 느림.
- 복구 시간은 incremental backup보다는 빠르지만 full backup보다는 느리다.

Backup의 종류 : Differential Backup

백업 방식	요일	백업 분량	복원 시 필요 파일
전체 Full	월	2G	월요일자 백업 파일 1개
차등 differential	화	1G	월요일자 백업 파일 1개와 화요일자 차등 백업 파일 1개
〃	수	2G	월요일자 백업 파일 1개와 수요일자 차등 백업 파일 1개
〃	목	3G	월요일자 백업 파일 1개와 목요일자 차등 백업 파일 1개
〃	금	4G	월요일자 백업 파일 1개와 금요일자 차등 백업 파일 1개
〃	토	5G	월요일자 백업 파일 1개와 토요일자 차등 백업 파일 1개
〃	일	6G	월요일자 백업 파일 1개와 일요일자 차등 백업 파일 1개
전체	월	9G	월요일자 백업 파일 1개
특징	전체 백업 파일과 마지막 차등 백업 파일 두 개만 필요하다.		

Backup의 종류: Mirror Backup

- 선택한 파일들을 압축없이 모든 것(시스템 구동에 필요한 것 포함)을 그대로 복사하는 백업 > 많은 프로그램을 설치했을 때 유용하다
- 백업 데이터의 정확한 복제본을 만들기 위해 종종 사용
- 백업 시간은 굉장히 빠르지만 많은 용량을 필요로 한다!

Backup의 종류 : 정리

	백업되는 데이터	백업 시간	복구 시간	백업 공간
Full	전부	매우 느림	빠름	높음
Incremental	마지막 Full Backup이후 변경점	빠름	보통	아주 낮음
Differential	마지막 Backup 이후 변경점	보통	빠름	보통
Mirror	모든 것(이미지 포함)	아주 빠름	아주 빠름	아주 높음

Backup을 위한 파일 묶기/압축

파일 아카이브 (with tar)

- **아카이브**란? 파일을 묶어서 하나로 만드는 것
- Tar 명령어: 여러 파일이나 디렉토리를 묶어서 하나의 파일로 만드는 명령어. ***압축을 하지는 않는다!***
- 압축을 하려면 파일을 묶으면서(tar) 동시에 압축(.gz or bz2)을 진행한다

파일 아카이브 (with tar, gz)

1. 백업용 디렉토리 만들기
   - $cd /
   - $sudo mkdir backups
   - $cd backups

2. **Tar만으로는 압축을 하진 않음!** 압축을 하려면 .gz나 bz2를 이용한다
   - $tar [-option] [tar파일(or gz)] --directory=<백업하고 싶은 디렉토리> --exclude=<제외할 항목> .
   - $tar -cvpf /backups/<백업파일이름>.tar --directory=/ --exclude=proc -exclude=sys -exclude=dev/pts --exclude=backups .
   - 주의: 뒤에 .을 생략하면
     ```
     tar: Cowardly refusing to create an empty archive
     Try 'tar --help' or 'tar --usage' for more information.
     ```
   - Option: -c 파일을 tar로 묶음, -v 과정을 화면으로 출력, -p 파일 권한을 저장, -f 파일 이름 지정
     -z gzip으로 압축하거나 해제, -j bzip2로 압축하거나 해제

파일 아카이브 (with tar, gz)

3. 복구하기
   - Tar xvpf <백업파일이름>.tar
   - Option: -x tar파일을 푼다

4. bz2, gz > 똑같은 방식으로 해줄 수 있다.
   - -z(gzip으로 압축하거나 해제) option을 사용.
   - Ex: $tar -zcvpf /backups/full-backup.tar.gz(or .bz2) --directory=/ --exclude=proc .

Backup 하기

dd(Data Duplicator)란?

- 블록 단위로 파일을 복사하거나 파일 변환을 할 수 있는 명령어
- 리눅스 시스템의 command라서 별도의 설치가 필요 없다!
- 데이터를 전송하거나 복구하는 용도로 사용할 수 있다.

- 주요 Option
  - if={입력대상}
  - of={출력대상}
  - bs={한 번에 읽고 쓸 최대 바이트 크기 지정}
  - count={지정한 블록 수 만큼 복사}

dd

- $dd if={복제할 대상이 되는 이름} of={복제본을 저장할 곳의 이름} bs={바이트} count={횟수}

- 예: /home/gony/a/hello.txt를 /home/gony/b/hi에 1024바이트 크기로 쓰는 것을 10번 복제.

  $dd if=/home/gony/a/hello.txt of=/home/gony/b/hi bs=1024 count=10

dd - 디스크 이미지 뜨기

- $ fdisk -l : 디스크 정보 가져오기

```
$sudo fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: ADBB86DA-BE1D-43DF-B6BA-55A6D3142FD5

Device     Start       End   Sectors Size Type
/dev/sda1   2048      4095      2048   1M BIOS boot
/dev/sda2   4096 41940991  41936896  20G Linux filesystem
```

$ dd if=backup.img of=/dev/sda2 bs=512 count=41936896 status=progress

Backup in remote server

- 같은 서버에 백업을 할 경우 서버에 문제가 생기면 날아간다!
  (ex: rm -rf /* -r: 파일 및 하위 디렉토리까지 삭제, -f: 강제로 삭제)

- 고로 다른 서버로 백업을 하는 게 좋다.

rsync 란?

- Remote Sync. 원격에 있는 파일과 디렉토리를 복사하고 동기화 하기 위해 사용하는 툴이며 동시에 네트워크 프로토콜.
- 리눅스에서는 백업 용도로 사용한다.

- $sudo apt install rsync
- $rsync <options> <source> <destination>
- 옵션
  - -v (--verbose) 자세한 정보 출력
  - -q (--quiet) 어떠한 메시지도 출력하지 않음(에러 포함)
  - -a (--archive) 원본 자체를 저장 관리(아카이빙). 위치, 권한, 소유주 포함하여 가져옴
  - -r (--recursive) 지정한 디렉토리의 하위 디렉토리까지 재귀적으로 실행
  - -z (--compress) 압축해서 전송
  - --progress: 진행 상태 표시

백업을 해보자! (with rsync and pem file)

EC2 인스턴스에 백업 하기

1. public key 생성
   - $ssh-keygen -t rsa
   - Enter를 누른다.
2. public key를 instance에 넣기
   - $scp -I ~/.ssh/<pem file 이름>.pem ~/.ssh/id_rsa.pub <사용자>@<주소>:~/.ssh/authorized_keys
3. 이 과정을 통해 local의 private key와 instance의 public key가 한 쌍이 된다.
4. 이 key를 이용해 데이터를 주고 받는다.
   - rsync -avz hello.txt <사용자>@<주소>:<백업할 곳>

or

$rsync -e 'ssh -I ~/.ssh/sparcs_seminar.pem' -avz hello.txt ubuntu@<주소>:/home/ubuntu/

주기적인 백업

- 1주차 때 배운 shell script와 cron을 이용하여 주기적으로 백업할 수 있다!
- 예)

Backup.sh

```
#!/usr/bin/env bash
Date=`date+%Y-%m-%d %H:%M:%S`
filename=backup_${DATE}.tar
tar -cvpf /backups/${filename} --directory=/ --exclude=proc -exclude=sys -exclude=dev/pts --exclude=backups
rsync -avz /backups/${filename} ubuntu@52.79.135.17:/home/ubuntu/

#매일 23시 59분에 백업
$crontab -e
59 23 * * * /home/gony/cron/Backup.sh
```

Emergency

Emergency (비상사태)

- Emergency가 발생할 수 있는 요인
  - Software Failure
  - Hardware Failure
  - By Human

Software Failure

- *파일 시스템* 에러
- 장치 설정 에러
- 부팅 에러
- 기타 프로그램 에러
- 커널 패닉(블루 스크린)
- 메모리 오버플로우

- 해킹
- 악성 코드, 바이러스
- 접속자 폭주

Hardware Failure

- 랜선 고장
- 전원장치 고장
- 파워 이상
- 냉각 이상
- 특정 부품/부분 망가짐

- 먼지
- 물 쏟음
- 케이블 절단

By Human

- 관리자 실수
- 악의적 내부자
- 잘못된 입력 혹은 오타

- 도둑
- 해커
- 악의적 사용자

파일 시스템은 언제 손상될까?

1. 시스템이 갑자기 중지, 전원이 나가버렸을 때.
   - 서버의 갑작스러운 종료로 인해 생겨난 시스템 이상으로, 재부팅시 파일 시스템이 마운트되지 않는 경우가 종종 발생 > 시스템을 부팅하여 사용하기 전에 fsck명령을 사용해 점검 필요
2. 두 프로세스가 인지하지 못한채 동시에 서로 내부 구조나 내용을 변경했을 때 (lock이 제대로 안된 상황일 때)
3. 한 프로세스에 의해 파일이 열려 있는데, 다른 프로세스가 그 파일을 삭제하려고 할 때
4. 시스템 디버거가 잘못 쓰였을 때

파일 시스템 점검 및 복구 – fsck

- 점검하고자 하는 디스크를 꼭 unmount 시킨 후 사용

  mount되어 있으면 디스크가 망가질 확률이 높다

  - umount {장치명}

- fsck {option} {장치명}

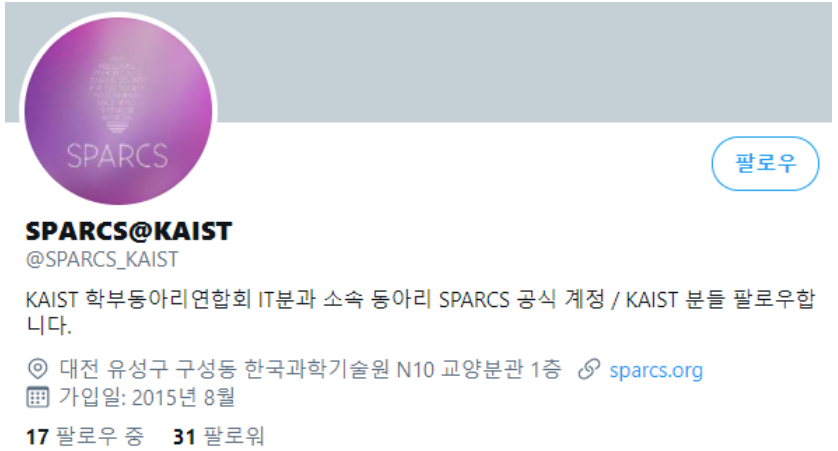
  -v: 자세한 출력

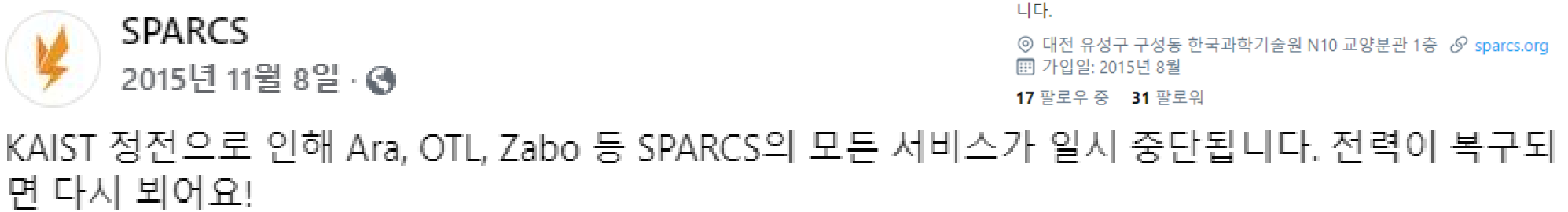
  -f: 파일 시스템 이상 유무와 상관없이 강제 체크

```
root@server:~# df
Filesystem     1K-blocks    Used Available Use% Mounted on
udev              978048       0    978048   0% /dev
tmpfs             201728    1188    200540   1% /run
/dev/sda2       20508240 4464624  14978812  23% /
tmpfs            1008640       0   1008640   0% /dev/shm
tmpfs               5120       0      5120   0% /run/lock
tmpfs            1008640       0   1008640   0% /sys/fs/cgroup
/dev/loop0         91648   91648         0 100% /snap/core/6130
/dev/loop1         89088   89088         0 100% /snap/core/4917
tmpfs             201728       0    201728   0% /run/user/0
root@server:~# fsck /dev/sdb1
fsck from util-linux 2.31.1
e2fsck 1.44.1 (24-Mar-2018)
/dev/sdb1: clean, 11/131072 files, 17964/524032 blocks
```

정전

- 정전 발생시 당황하지 않기
- 서버실에는 UPS(무정전전원공급장치)가 있어서 어느 정도 시간의 서버들이 멈추지 않고 작동할 수 있다!
- 서버를 종료하기 전에 SPARCS로 서비스 중단 공고(sns, email 등)를 하고, 서버들을 차례대로 종료하도록 한다.





해킹

- 해킹을 당하면 대부분 명령어를 못쓰게 하거나 중요 파일 삭제, 변조를 일으킨다.
- 미리 백업해둔 시스템 코어로 대체하여 해결하자!
- 해킹 사례: 2012 KAIST/SPARCS 해킹 사건

  https://s3.ap-northeast-2.amazonaws.com/sparcs.home/seminars/chocho-20140806_1-0.pdf 35~47page
- 예방
  - 안쓰는 port 닫기 > ec2 instance에서 '위치무관' 무분별 사용 X
  - 의심가는 process 죽이기
  - 주기적으로 프로그램 update
  - 모의 해킹 등으로 보안 점검
  - Root로 로그인하고 자리 비우지 않기
  - 바이러스 감염되지 않도록 주의
  - 서버실 물리적 보안

서버실의 온도

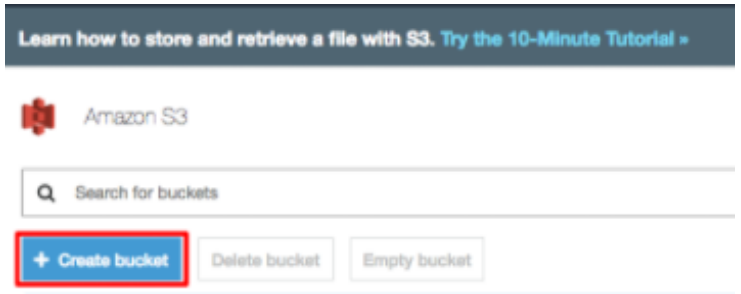
- 높은 온도는 고장의 원인
- 서버실의 온도는 겨울철 26도, 여름철 33~37도로 높은 편.
- 서버실에 온도계가 설치되어 있으니 자주 확인한다.
- 서버실 온도가 높다 > 에어컨의 작동여부 확인 > 고장난 경우 리모컨으로 전원을 다시 켜보고 온도를 최저로 낮춘다.
- 가끔 멀티탭에 문제가 있는 경우도 있으니 확인한다.

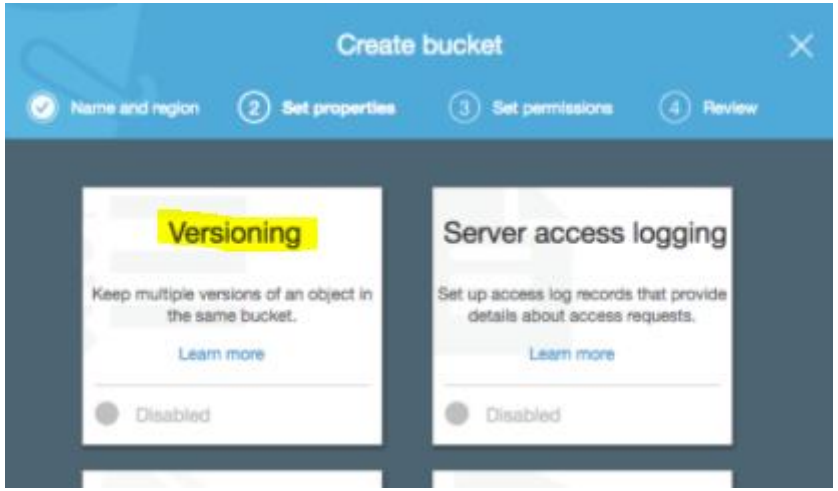
  >2009년 서버실의 메인콘센트 유체이탈(by 에어컨의 엄청난 소비전력)
- 에어컨에 문제가 있다면 시설팀에 연락하여 수리 요청.
- 서버실의 문을 열고, 선풍기 등을 이용해 열을 빼낸다. (단, 보안 신경쓰기)
- 비상시, 중요하지 않은 서버들을 종료하여 최대한 발열량을 줄인다.

AWS에서 안전하게 사용하기

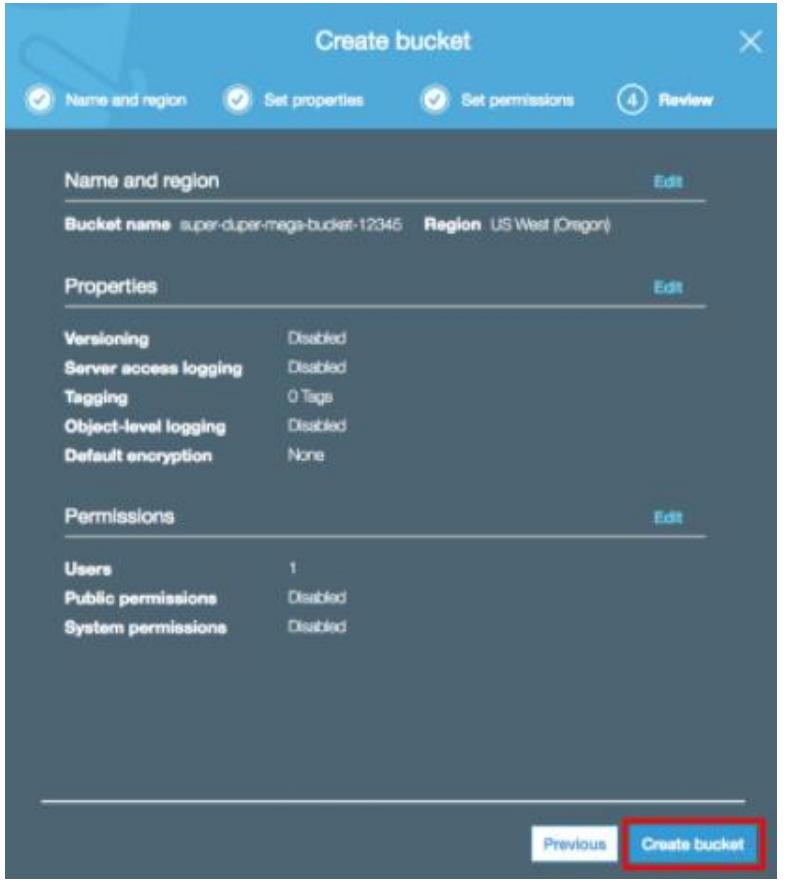
AWS에서 안전하게 사용하기 (with S3)

- S3(Simple Storage Service): 객체 스토리지 서비스
- S3 버킷을 생성하여 버킷에 중요한 파일을 업로드한다.

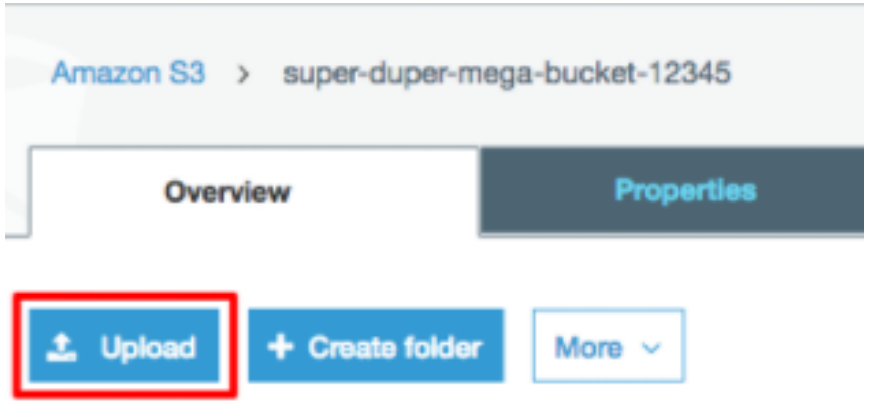


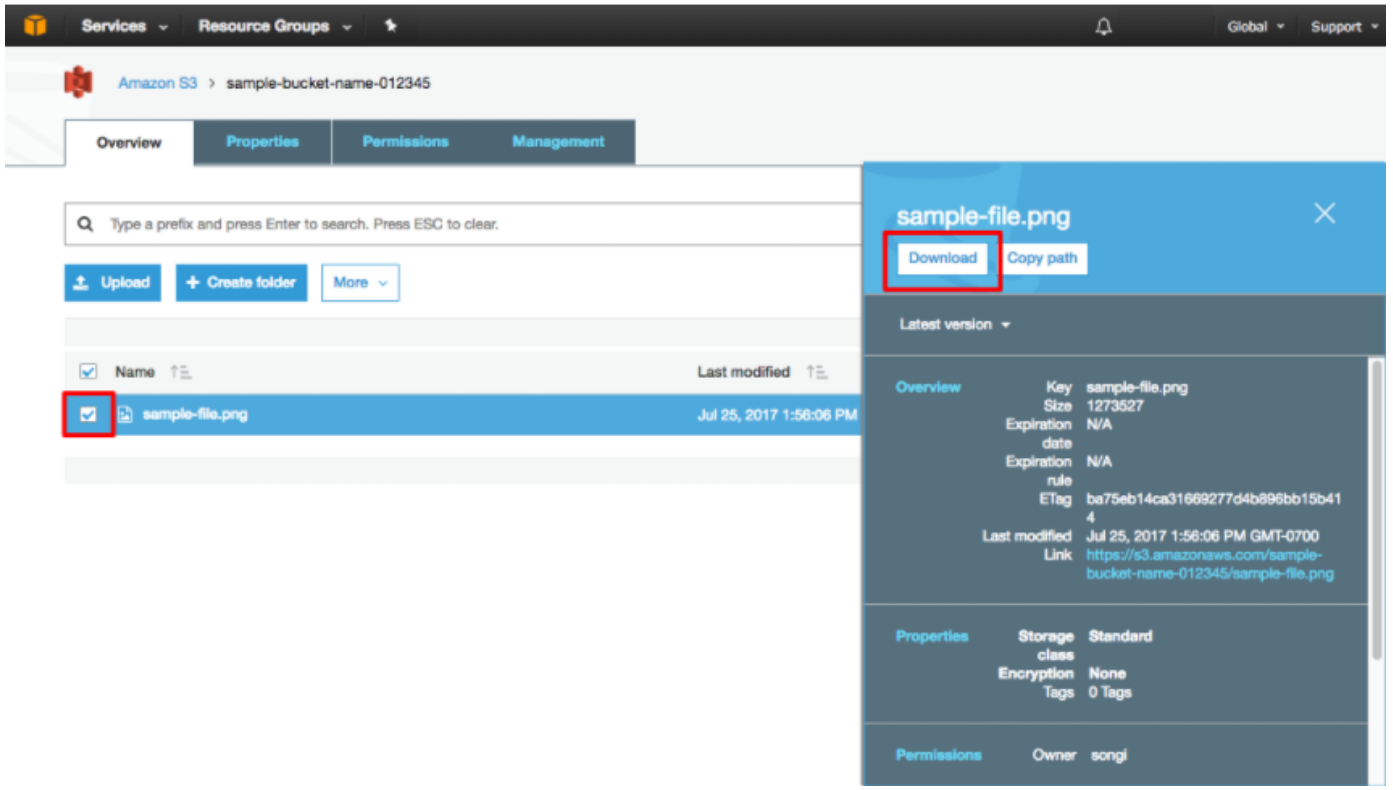


버전 관리를 통해 한 파일의 수정내역에 따른 버전을 보관할 수 있다!



AWS에서 안전하게 사용하기 (with S3)





S3를 이용하여 보다 안전하게 파일을 관리할 수 있다!

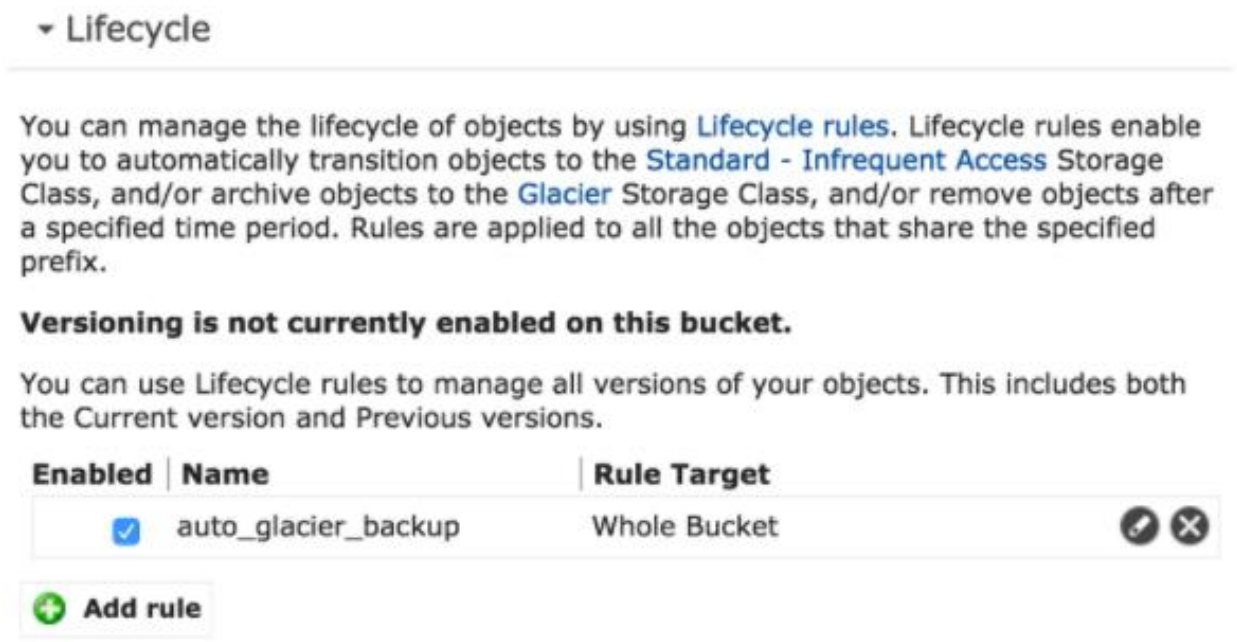
AWS에서 안전하게 사용하기 (with S3 glacier)

- S3 glacier: 데이터 보관 및 장기 백업을 위한 안전하고 안정적이며 비용이 매우 저렴한 Amazon S3 스토리지 클래스
- Glacier은 언제 사용할까?
- 예: 방송국 창고의 무수한 테이프들. 평소에는 굳이 찾아볼 필요가 없다. 하지만 백업이 필요한 특정한 상황일 때 필요!
- 이러한 경우에 유용하다. 왜냐하면 비용은 최소화하며 반영구적으로 안전하게 저장하기 때문

AWS에서 안전하게 사용하기 (with S3 glacier)

- 데이터를 장기적으로 보관한다면? S3보다 S3 glacier이 가격적인 측면에서 유리(서울기준 glacie는 s3대비 비용 약 1/3, s3와 동일한 내구성 99.999999999% 보장)
- S3에 있는 자료를 Glacier로 주기적으로 보내주자
- S3에서 Glacier로 자동 백업하는 법: bucket>properties>Lifecycle>Add rule>Rule targe지정>Rule 설정(Archive to the Glacier storage class)
- 출처: https://bluese05.tistory.com/39

Rule이 추가된 모습 >



AWS에서 안전하게 사용하기 (with S3 glacier)

유의점

- 마음대로 검색 제한
  - 검색의 경우 매달 최대 용량의 5%에만 무료로 허용 > 초과시 요금 부과
- 마음대로 삭제 제한
  - 기본 삭제 조건은 3개월 이상 저장시 > 조기 삭제시 요금 부과
- 아카이브 데이터 복구하는 데 오랜 시간 필요

-> 따라서 Glacier만 사용하는 것 보다는 최근 백업 데이터는 S3에 먼저 저장하고 오래된 백업 데이터는 자동으로 Glacier로 옮기는게 좋다!

AWS에서 안전하게 사용하기 (with RDS)

- RDS(Relational Database Service): AWS 클라우드에서 관계형 데이터베이스를 더 쉽게 설치, 운영 및 확장할 수 있는 웹 서비스
- Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle 및 Microsoft SQL Server DB 엔진 지원
- 간단한 사용법
- 간편한 백업 - 스냅샷 기능으로 백업 가능
- 스토리지 및 iops(단위 시간당 읽기/쓰기 횟수)확장이 용이

- 생성 및 이용법 https://hiroki-yim.github.io/study/aws/rds/running-aws-rds-post/#rds%EB%9E%80

AWS에서 안전하게 사용하기 (with RDS)

- RDS 사용 vs EC2에 DB를 직접 설치
- AWS에 지불하는 비용: 전자 > 후자
- 기본적인 관리, DB최적화도 알아서 해줌 > 관리적인 측면에서 이점
- 관리 측면의 이득 > AWS에 지불하는 비용 -> RDS선택이 좋다!

AWS에서 안전하게 사용하기 (with RDS)

- 자동 백업: RDS는 DB 인스턴스 백업 기간 동안 전체 DB 인스턴스의 자동 백업하여 DB 인스턴스의 스토리지 볼륨 스냅샷을 생성
- 기본 백업 기간 동안 매일 실행: 서울의 경우 13:00-21:00 UTC(새벽)
- 백업 보존 기간: 1일 ~ 35일 설정 가능, default: 1일

- https://docs.aws.amazon.com/ko_kr/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html

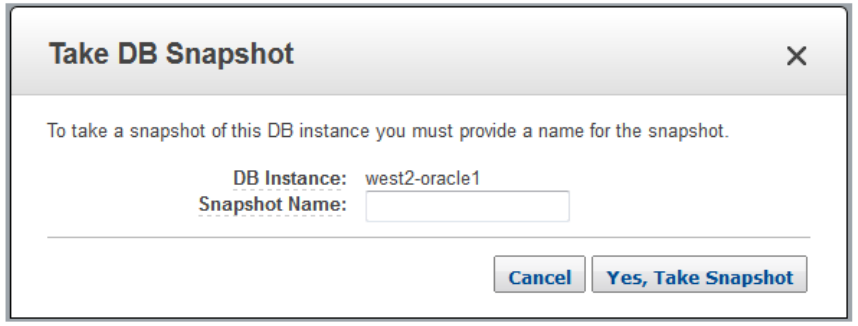
AWS에서 안전하게 사용하기 (with RDS)

- 수동 백업: 수동으로 DB 스냅샷을 생성하여 DB 인스턴스를 백업

DB 스냅샷을 생성하려면

1. AWS Management 콘솔에 로그인한 후 https://console.aws.amazon.com/rds/ 에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 **데이터베이스**를 선택합니다.
3. DB 인스턴스 목록에서 스냅샷을 생성하려는 DB 인스턴스를 선택합니다.
4. **작업**에서 **스냅샷 만들기**를 선택합니다.

   [Take DB Snapshot] 창이 나타납니다.
5. **스냅샷 이름** 상자에 스냅샷의 이름을 입력합니다.



6. **Take Snapshot(스냅샷 생성)**을 선택합니다.

DB 스냅샷에서 DB 인스턴스를 복원하려면

1. AWS Management 콘솔에 로그인한 후 https://console.aws.amazon.com/rds/ 에서 Amazon RDS 콘솔을 엽니다.
2. 탐색 창에서 **[Snapshots]**를 선택합니다.
3. 복원 원본으로 사용할 DB 스냅샷을 선택합니다.
4. **작업**에서 **스냅샷 복원**을 선택합니다.
5. **DB 인스턴스 복원** 페이지의 **DB 인스턴스 식별자**에 복원된 DB 인스턴스의 이름을 입력합니다.
6. **[Restore DB Instance]**를 선택합니다.

Convenience

IaC (Infrastructure as code)

- 인프라 구성을 마치 소프트웨어를 프로그래밍하는 것처럼 처리하는 방식

- 반복작업을 자동화하여 작업 시간 단축과 사람에 의한 에러 감소, 빠른 복원이 특징
- 예) 유사한 설정을 가진 여러 인스턴스 생성시

- 가상 시스템에 대한 프로그래밍 방식의 관리를 지원 > 개별 하드웨어를 수동으로 구성하고 업데이트 할 필요가 없다!
- 예) 인스턴스의 type을 small에서 large로 변경시 AWS 콘솔 웹에서 일일이 변경하지 않고 code로 간단히 변경

IaC (Infrastructure as code) - Terraform

- HashiCrop에서 만든 오픈소스 인프라 관리 도구
- $ brew install terraform
- HCL(Hashicorp Configuration Language) 설정 언어 사용. 확장자(.tf)
- 코드 작성 후 $ terraform apply
- 1분 내로 EC2 인스턴스 생성!

- https://www.44bits.io/ko/post/terraform_introduction_infrastrucute_as_code

누구나 알고 있지만 한 번 더 되새기자

‘한번의 백업이 하룻밤 삽질, 10번의 복구작업,
100번의 후회와 해고를 막습니다’

by sillo 선배님

Thank you!